

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



**Grado en Ingeniería Informática**

**TRABAJO FIN DE GRADO**

**GESTOR DE PROYECTOS**

**Miguel José Moratilla López**  
**Tutor: Eduardo Salas Gómez**  
**Ponente: Iván González Martínez**

**MARZO 2019**



# **GESTOR DE PROYECTOS**

**AUTOR: Miguel José Moratilla López**

**TUTOR: Eduardo Salas Gómez**

**Dpto. Ingeniería Informática  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Marzo 2019**



# Resumen

Este Trabajo Fin de Grado consiste en el desarrollo de una aplicación web que tiene como objetivo facilitar la organización y la gestión del trabajo de los empleados de una empresa.

Organiza el trabajo en proyectos, tareas y equipos formados por miembros, permitiendo visualizar el estado del proyecto mediante diagramas y calendarios, pudiendo ver gráficas relacionadas con los empleados o el estado de los proyectos, además de informes.

Se ha añadido también la posibilidad de realizar reuniones de proyecto para tener una mejor organización del equipo en el mismo.

En la aplicación Web los empleados, según el trabajo que desempeñen, se les asignará el papel de: administrador, líder o miembro del equipo del proyecto. Además, existe la posibilidad de incluir colaboradores externos en el equipo.

Se basa en la arquitectura modelo-vista-controlador mediante el framework ASP.NET MVC y se ha utilizado el modelo de ciclo de vida en cascada para la realización del proyecto.

Este proyecto se centra en conseguir una aplicación sencilla y estética que llegue a todos los usuarios de la empresa y que pueda ser ejecutada en distintos tipos de dispositivos gracias al uso de tecnologías como Bootstrap, que permite una interfaz completamente responsive.

Por último, se han desarrollado distintos tipos de pruebas para comprobar el correcto funcionamiento y la robustez de la aplicación.

# Abstract

This Final Project consists on the development of a web application that aims to facilitate the organization and management of the work, that employees produce in a company.

It organizes the work in projects, tasks and teams formed by members, allowing it to visualize the status of the project through diagrams and calendars, also provides the ability to see a wide range of graphs related to the members or status of the projects, as well as reports.

Furthermore, it includes the possibility to hold project meetings and therefore a better organization of the team itself

In the Web application, employees, depending on the work they perform, will be assigned the role of: administrator, leader or member of the project team. In addition, there is the possibility of including external collaborators as part of the team too.

It is based on the model-view-controller architecture using the ASP.NET MVC framework and the development methodology used has been the waterfall.

This project focuses on getting a simple and yet visual application that reaches all users of a company and can be executed on different types of devices, thanks to the use of technologies such as Bootstrap which allows a completely responsive interface.

Finally, different types of tests have been developed to verify the correct operation and robustness of the application.

## **Palabras clave**

Aplicación Web, Framework, Bootstrap, metodología, cascada, responsive, mvc, .NET, Visual Studio, c#, procedimientos almacenados

## **Keywords**

Web Application, Framework, Bootstrap, methodology, waterfall, responsive, mvc, .NET, Visual Studio, c#, stored procedures

## *Agradecimientos*

*A todos los profesores de la escuela que me han impartido clases y tutorías y se han preocupado por mí a lo largo de estos años. Destacando entre ellos Iván González, Idoia Alarcón y Eloy Anguiano, por apoyarme este último año y darme siempre consejo cuando lo he necesitado.*

*A todos los compañeros que he conocido durante todos estos años, estudiando juntos en la biblioteca y salas de estudio, haciéndome más fácil resolver dudas y superar las asignaturas más difíciles.*

*Por último, a mi familia, en especial a mi madre y a mi novia, las cuales han sido los pilares de apoyo más importantes, permitiéndome llegar hasta aquí y superar momentos muy difíciles.*





## INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	1
1.3	Organización de la memoria.....	2
2	Estado del arte .....	3
2.1	Alternativas actuales .....	3
2.1.1	Tabla comparativa .....	4
2.2	Estudio de tecnologías .....	5
2.2.1	Framework .net.....	5
2.2.1.1	Tabla comparativa .....	5
2.2.2	Framework Bootstrap .....	7
2.2.3	jQuery .....	7
2.2.4	AJAX.....	8
3	Modelo de ciclo de vida, Análisis y Diseño .....	9
3.1	Modelo de ciclo de vida.....	9
3.2	Análisis .....	10
3.2.1	Especificación de requisitos .....	10
3.2.1.1	Requisitos funcionales .....	10
3.2.1.2	Requisitos no funcionales .....	13
3.2.2	Diagrama de casos de Uso.....	14
3.2.3	Modelo de datos.....	15
3.3	Diseño.....	16
3.3.1	Diseño de la arquitectura .....	17
3.3.2	Diseño de las tablas .....	18
4	Desarrollo .....	19
4.1	Hardware .....	19
4.2	Software.....	19
4.2.1	Entornos de desarrollo .....	20
4.2.2	Lenguajes de programación.....	20
4.2.3	Flujo de datos.....	24
4.2.4	Uso de código externo: Plugins .....	26
4.2.5	Conceptos de desarrollo.....	29
5	Integración, pruebas y resultados .....	33
5.1	Pruebas Unitarias .....	33
5.2	Pruebas Funcionales .....	33
5.3	Pruebas Integración .....	36
5.4	Pruebas Regresión .....	36
6	Conclusiones y trabajo futuro.....	37
6.1	Conclusiones.....	37
6.2	Trabajo futuro .....	37
	Referencias .....	39
	Glosario .....	41
	Anexos.....	I
A	Anexo.....	I
i.	Anexo Especificación de requisitos.....	I
ii.	Anexo Pruebas Funcionales.....	III
iii.	Ficheros de configuración .....	VI

iv.	Organización de las carpetas .....	VII
-----	------------------------------------	-----

## INDICE DE FIGURAS

FIGURA 1: FORMULARIO CON BOOTSTRAP .....	7
FIGURA 2: MODELO DE CICLO DE VIDA EN CASCADA .....	9
FIGURA 3: DIAGRAMA DE CASOS DE USO .....	15
FIGURA 4: MODELO ENTIDAD-RELACIÓN.....	16
FIGURA 5: M-V-C .....	18
FIGURA 6: DISEÑO BASE DE DATOS .....	18
FIGURA 7: EJEMPLO CSS.....	21
FIGURA 8: EJEMPLO TRABAJO .....	23
FIGURA 9: LLAMADA AJAX.....	24
FIGURA 10: LLAMADA DESDE CONTROLADOR AL CONTROLADOR GESTOR_SP .....	25
FIGURA 11: LLAMADA AL PROCEDIMIENTO ALMACENADO .....	25
FIGURA 12: FUNCIÓN EXECUTE CADENA .....	26
FIGURA 13: PROCEDIMIENTO ALMACENADO .....	26
FIGURA 14: INSERTAR FECHA .....	27
FIGURA 15: FUNCIÓN CALENDARIO .....	27
FIGURA 16: FUNCIÓN BARRA PROGRESO .....	27
FIGURA 17: VISTA PROYECTO .....	28
FIGURA 18: DIAGRAMA DE TAREAS DEL PROYECTO.....	28
FIGURA 19: GRÁFICA PROYECTO-USUARIO.....	29
FIGURA 20: GRÁFICA DIARIA-USUARIO .....	29
FIGURA 21: EJEMPLO HTML DINÁMICO PARA FILTRAR USUARIOS.....	30
FIGURA 22: EJEMPLO GRID DEL FILTRO DE PROYECTOS.....	31

# INDICE DE TABLAS

<b>TABLA 1: HERRAMIENTAS ALTERNATIVAS .....</b>	<b>4</b>
<b>TABLA 2: COMPARACIÓN .NET Y PHP.....</b>	<b>6</b>
<b>TABLA 3: REQUISITO FUNCIONAL 1 .....</b>	<b>10</b>
<b>TABLA 4: REQUISITO FUNCIONAL 2 .....</b>	<b>11</b>
<b>TABLA 5: REQUISITO FUNCIONAL 3 .....</b>	<b>11</b>
<b>TABLA 6: REQUISITO FUNCIONAL 4 .....</b>	<b>11</b>
<b>TABLA 7: REQUISITO FUNCIONAL 5 .....</b>	<b>12</b>
<b>TABLA 8: REQUISITO FUNCIONAL 6 .....</b>	<b>12</b>
<b>TABLA 9: REQUISITO FUNCIONAL 7 .....</b>	<b>12</b>
<b>TABLA 10: REQUISITO FUNCIONAL 8 .....</b>	<b>13</b>
<b>TABLA 11: PRUEBA RF1 .....</b>	<b>33</b>
<b>TABLA 12: PRUEBA RF2 .....</b>	<b>33</b>
<b>TABLA 13: PRUEBA RF3 .....</b>	<b>34</b>
<b>TABLA 14: PRUEBA RF4 .....</b>	<b>34</b>
<b>TABLA 15: PRUEBA RF5 .....</b>	<b>34</b>
<b>TABLA 16: PRUEBA RF6 .....</b>	<b>34</b>
<b>TABLA 17: PRUEBA RF7 .....</b>	<b>35</b>
<b>TABLA 18: PRUEBA RF8 .....</b>	<b>35</b>
<b>TABLA 19: PRUEBA RF9 .....</b>	<b>35</b>
<b>TABLA 20: REQUISITO FUNCIONAL 9 .....</b>	<b>I</b>
<b>TABLA 21: REQUISITO FUNCIONAL 10 .....</b>	<b>I</b>
<b>TABLA 22: REQUISITO FUNCIONAL 11 .....</b>	<b>II</b>
<b>TABLA 23: REQUISITO FUNCIONAL 12 .....</b>	<b>II</b>
<b>TABLA 24: REQUISITO FUNCIONAL 13 .....</b>	<b>II</b>
<b>TABLA 25: REQUISITO FUNCIONAL 14 .....</b>	<b>III</b>

<b>TABLA 26: REQUISITO FUNCIONAL 15 .....</b>	<b>III</b>
<b>TABLA 27: PRUEBA RF17 .....</b>	<b>III</b>
<b>TABLA 28: PRUEBA RF18 .....</b>	<b>IV</b>
<b>TABLA 29: PRUEBA RF19 .....</b>	<b>IV</b>
<b>TABLA 30: PRUEBA RF10 .....</b>	<b>IV</b>
<b>TABLA 31: PRUEBA RF11 .....</b>	<b>IV</b>
<b>TABLA 32: PRUEBA RF12 .....</b>	<b>V</b>
<b>TABLA 33: PRUEBA RF13 .....</b>	<b>V</b>
<b>TABLA 34: PRUEBA RF14 .....</b>	<b>V</b>
<b>TABLA 35: PRUEBA RF15 .....</b>	<b>V</b>

# 1 Introducción

---

En este capítulo se explica la motivación del TFG, siendo este un desarrollo software de gestión perteneciente al ámbito de la Ingeniería del Software, detallando cuales han sido los objetivos y como se ha organizado esta memoria.

## 1.1 Motivación

Este trabajo de fin de grado está dirigido a una empresa con el objetivo de conseguir una organización correcta de sus proyectos y la contabilización de las horas de trabajo diario por parte de cada empleado. Con ese fin, se pretende mejorar lo que hasta ese momento era ineficiente y suponía un retraso a la hora de llevar a cabo el desarrollo de los proyectos debido a que no quedaba constancia del trabajo realizado ni del estado de los mismos.

Por ello, se ha decidido implementar una aplicación web, de tal forma que, mediante cualquier dispositivo, puedan reflejar el trabajo realizado durante el día, pudiendo así los supervisores y la dirección visualizarlo y realizar cambios en la organización de cada proyecto si fuera necesario y tomar las medidas oportunas.

La aplicación web gestiona la creación de proyectos con sus características, además de sus tareas, equipo, miembros, líder de equipo, colaboraciones externas y reuniones, pudiendo llevar de forma más ordenada un seguimiento total.

Por tanto, se ha propuesto esta aplicación web como TFG con el nombre de **Gestor de Proyectos**.

## 1.2 Objetivos

El objetivo principal del Trabajo de Fin de Grado será desarrollar una aplicación web que sirva como herramienta de administración y consulta de proyectos y empleados.

Para lograr cumplir con los requisitos del sistema la herramienta debe permitir:

- El registro en la aplicación con distintos tipos de usuarios: líder de equipo de proyecto, miembro del equipo del proyecto y colaborador externo.
- Ofrecer un perfil de usuario con diversa información.
- La creación de proyectos con su equipo de desarrollo, sus respectivas tareas y tener la posibilidad de listarlos.
- La creación de una vista que ofrezca una interfaz de usuario clara e intuitiva de utilizar, que no provoque errores de uso (se ofrece un manual de usuario accesible desde la aplicación).
- La creación de calendarios y diagramas para visualizar tareas y reuniones.
- El almacenamiento en el servidor de informes relacionados con estadísticas de los proyectos.
- Ofrecer un sistema de mensajería interno a la aplicación.
- Ofrecer gráficas y estadísticas relacionadas con los proyectos y empleados.

- Crear el modelo de la base de datos que permita guardar y consultar todos los proyectos.

### **1.3 Organización de la memoria**

La memoria consta de los siguientes capítulos:

- **Capítulo 2: Estado del Arte.** En este capítulo se analizan otras alternativas al desarrollo mediante una tabla comparativa, en concreto una de ellas fue utilizada a modo de prueba por la empresa. Además, se hablará de las distintas tecnologías utilizadas, el porqué de su uso y se añadirá otra tabla comparativa de las tecnologías usadas frente a otras posibles.
- **Capítulo 3: Modelo de ciclo de vida, Análisis y Diseño.** En este capítulo se define el modelo de ciclo de vida utilizado, el análisis de los requisitos funcionales y no funcionales de la aplicación, el diagrama de casos de uso, el modelo de datos, el diseño de la arquitectura y el diseño de la base de datos.
- **Capítulo 4: Desarrollo.** En este capítulo se describirán el hardware, los entornos de desarrollo y lenguajes utilizados para el desarrollo, además de hablar del flujo de datos que sigue la aplicación y algunos conceptos y ejemplos del desarrollo.
- **Capítulo 5: Integración, Pruebas, Resultados.** En este capítulo se detallan las pruebas realizadas para comprobar el correcto funcionamiento de la aplicación.
- **Capítulo 6: Conclusión y trabajo futuro.** En este último capítulo se exponen las conclusiones y reflexiones obtenidas antes, durante y tras el desarrollo de la aplicación, así como posibles mejoras o ideas de implementación para el futuro.
- **Tras estos capítulos tendremos la bibliografía, glosario y los apéndices del documento.**

## 2 Estado del arte

---

En este capítulo se exponen distintas posibilidades de gestión de proyectos mediante herramientas ya existentes en el mercado, que puedan solucionar el problema expuesto en los apartados de la motivación y objetivos. Se comparan todas ellas y se sacan algunas conclusiones.

En el segundo punto se llevará a cabo el estudio de las tecnologías utilizadas y el porqué de su uso.

### 2.1 Alternativas actuales

Existen diversas herramientas que facilitan la gestión y organización de proyectos. Por ello, se estudian las que más se asemejan a los requisitos solicitados por la empresa y se analiza la herramienta sobre la que la empresa se inspiró para solicitar sus requisitos y que fue usada temporalmente a modo de prueba.

En total se analizan cuatro herramientas, la primera de ellas es Kanban Flow [\[1\]](#), llegó a ser utilizada en la empresa durante un breve período de tiempo, pero al final, se optó por un desarrollo propio para hacerla a medida y no tener problemas con los usuarios, ya que en la versión gratuita utilizada había un número máximo de usuarios en la aplicación y por ello, no servía para el uso que se le quería dar. Además de esto, se prefirió un desarrollo propio para poder integrar la aplicación con otros desarrollos internos de la empresa, como por ejemplo el módulo de incidencias que se usa para sacar las estadísticas diarias de cada empleado.

Las otras tres herramientas son Trello [\[4\]](#), Slack [\[4\]](#) y Wrike [\[2\]](#). Aparte de estas, hay muchas otras como [\[3\]](#): Assembla, Confluence, Central Desktop, Kapost, TeamBox, Teamlab o Basecamp, pero solo se analizan la que utilizó la empresa y otras que fueron planteadas en su momento.

**1. Kamban Flow:** es una herramienta de gestión de proyectos basada en la organización. Es útil para trabajo unipersonal pero también para trabajar con colaboradores en equipo y cuenta con versión para dispositivos móviles.

Para usarla, solo debes registrarte y crear el tablero, después se organizan columnas, las cuales se dividen en las tareas pendientes por hacer, las tareas que hacer hoy, las tareas en proceso de ejecución y las columnas en las que agregaremos las tareas completadas.

Podremos realizar acciones como; editar la descripción, el color, los miembros, añadir subtareas, dejar un comentario, acceder a los informes de resultados o borrar la tarjeta si quieres eliminarla.

Por último, se pueden ver informes con las estadísticas y datos (seleccionando periodo de días a ver, un tablero específico o un informe global).

**2. Trello:** es una herramienta muy fácil e intuitiva con un diseño muy visual que permite un seguimiento de la evolución de un proyecto pudiendo marcar las tareas terminadas, en proceso o bloqueadas. Está organizada en forma de tableros que permiten añadir anotaciones y adjuntar archivos.

**3. Slack:** es una herramienta más orientada a conseguir una muy buena comunicación del equipo de trabajo ahorrándonos enviar una gran cantidad de mensajes. Este sistema de mensajería también permite todo tipo de archivos ya sean de fragmentos de código o documentos.

**4. Wrike:** herramienta que combina la gestión de proyectos con un espacio de trabajo en tiempo real en los que los equipos colaboran, se comunican y comparten documentos. Todo ello permite estar al tanto del trabajo de los compañeros. Además, tiene una buena gestión de tareas con temporizadores visuales y con un buen sistema de priorización de las mismas.

### 2.1.1 Tabla comparativa

Para desarrollar la aplicación se ha intentado extraer lo mejor de cada una de las herramientas intentando hacer el Gestor de Proyectos lo más completo y sencillo posible, evitando las desventajas que estas mismas presentan.

				
<b>Ventajas</b>	<ul style="list-style-type: none"> <li>- Fácil e intuitivo. Muy buena para el trabajo y coordinación en equipo.</li> <li>- Muy buena en la organización de proyectos.</li> </ul>	<ul style="list-style-type: none"> <li>- Fácil de instalar y manejo intuitivo.</li> <li>- Aplicaciones móviles para IOS y Android.</li> <li>- Diseño personalizable con muchas opciones para elaborar y categorizar tareas.</li> </ul>	<ul style="list-style-type: none"> <li>- Facilidad para compartir archivos, código y documentos.</li> <li>- Aplicaciones para IOS, Android y Windows.</li> <li>- Diferentes interfaces a servicios externos.</li> </ul>	<ul style="list-style-type: none"> <li>- Función de copiado por tareas y proyectos repetidos.</li> <li>- Aplicaciones para IOS, Android y Windows.</li> <li>- Actividades en Stream y control de versiones.</li> </ul>
<b>Desventajas</b>	<ul style="list-style-type: none"> <li>- No tiene sistema de mensajería ni chat en tiempo real.</li> <li>- Relativamente caro.</li> <li>- Permite solo 3 usuarios en su versión gratuita.</li> </ul>	<ul style="list-style-type: none"> <li>- No se puede cambiar la posición y estructuras de las tarjetas de tareas en el tablero.</li> <li>- Archivos almacenados en servidores americanos.</li> </ul>	<ul style="list-style-type: none"> <li>- Interfaz de usuario en inglés.</li> <li>- Sin opción para crear equipos.</li> <li>- Archivos almacenados en servidores americanos.</li> </ul>	<ul style="list-style-type: none"> <li>- Relativamente caro.</li> <li>- Interfaz de usuario poco intuitiva.</li> <li>- Archivos almacenados en servidores americanos.</li> </ul>

**Tabla 1: Herramientas alternativas**



## 2.2 Estudio de tecnologías

En este punto se realiza un estudio de las tecnologías que se han utilizado para el desarrollo del Gestor de Proyectos.

### 2.2.1 Framework .net

Framework .net [\[6\]](#) es un framework de Microsoft centrado en la transparencia de redes, con independencia de plataforma de hardware que permite un rápido desarrollo de aplicaciones. Este framework surge como respuesta de Microsoft ante el creciente mercado de los negocios en entornos Web, como competencia a los diversos framework de desarrollo web basados en PHP. Ofrece una manera rápida y segura, de desarrollar aplicaciones permitiendo una rápida integración entre empresas y un simple y rápido acceso a todo tipo de información desde cualquier tipo de dispositivo.

El lenguaje C# es uno de los lenguajes desarrollados para el marco de trabajo .NET y es el utilizado en el Gestor de Proyectos.

**El ASP.NET MVC Framework** es un framework de aplicaciones web que implementa el patrón modelo-vista-controlador (MVC) anteriormente comentado.

#### 2.2.1.1 Tabla comparativa

A la hora de desarrollar el Gestor de Proyectos, lo primero que pensé fue utilizar PHP o alguno de sus frameworks para su desarrollo, debido a que fue un lenguaje estudiado en la escuela.

Por otro lado, existen muchos frameworks/librerías JavaScript que nos ayudan a desarrollar aplicaciones web de forma rápida y eficiente y que hubieran sido una buena opción en vez de .NET. Las más destacadas son Angular, React y Vuejs [\[27\]](#), las cuales, si te encuentras desarrollando una app con un alto nivel de interacción, son muy recomendables con el fin de evitar escribir muchas de las líneas de código que se necesitarían usando JQuery.

Sin embargo, se utiliza .NET para el Gestor de Proyectos dado que en la empresa se usa este framework para otros desarrollos y ya existe una infraestructura montada para ello.

A continuación, se muestra una comparativa entre PHP y .NET para sacar algunas conclusiones.

		
<b>Coste</b>	Sistema operativo windows.	Cualquier sistema operativo.
<b>Rendimiento</b>	C# tiene un mejor procesamiento en paralelo.	Mejor rendimiento gracias al Sistema EXT4 de linux, que es mas veloz que el NTFS de Windows.
<b>Seguridad</b>	Buena, pero tardan más en publicar parches y mejoras.	Buena, al ser código abierto la comunidad lo mantiene bien.
<b>Sintaxis</b>	Orientada a objetos. Tiene que ser realizada en Visual Studio.	No es orientada a objetos hasta la versión 5. Puede ser realizada en cualquier editor de texto.
<b>Portabilidad</b>	Como es de Microsoft tiene problemas de compatibilidad con el resto de sistemas operativos.	Buena portabilidad al ser multiplataforma.
<b>Tiempo de desarrollo</b>	Necesita compilar el código cada vez que se modifica.	Cuesta menos lograr objetivos más complejos, además se interpreta directo desde el servidor mismo.
<b>Flexibilidad</b>	Ampliamente usado por Microsoft.	Mas limitado al mundo web.
<b>Soporte y ayuda</b>	Buen soporte y ayuda pero al ser de Microsoft no tiene una comunidad tan extensa.	Buen soporte y ayuda al ser de código abierto.

**Tabla 2: Comparación .net y php**

Tras ver la comparativa obtenida de [\[7\]](#), seguramente hubiera utilizado PHP por las ventajas que presenta, entre ellas, el hecho de que estoy realizando un desarrollo web de gestión, además de ser el lenguaje estudiado en la universidad. Sin embargo, el haber usado .NET ha tenido otras ventajas, como mi aprendizaje en estas tecnologías, además de resultar igualmente eficiente.

## 2.2.2 Framework Bootstrap

Bootstrap [\[8\]](#) es un framework desarrollado que tiene como objetivo facilitar el diseño web. Permite crear fácilmente webs de diseño adaptable que se ajusten a cualquier dispositivo y tamaño de pantalla.

Se consigue que la carga de la web sea rápida y que la navegación sea fluida e intuitiva, facilitando mucho la construcción de una página.

Además, es posible utilizar elementos web tales como iconos o desplegados, combinando HTML, CSS y JavaScript ya que se integra muy bien con las principales librerías de esta.

En cuanto a desventajas que he encontrado, destacar que te tienes que adaptar al modelo de 12 columnas y que a la hora de mantenerlo o ampliarlo es más complicado y tedioso ya que si quieres añadir componentes que no existen, debes hacerlos tú mismo manteniendo la coherencia con tu diseño, cuidando el responsive.

A continuación, se muestra un ejemplo del uso de Bootstrap en el Gestor de Proyectos, haciendo uso de las 12 columnas en la clase del elemento HTML. Además, se usa la clase “row” para las filas.

```
<div class="row">
  <div class="col-md-10 col-sm-6 col-xs-12">
    <label class="textoBasico" id="textoNombreFiltroProyecto">Nombre</label>
    <br />
    <input style="width:79%" type="text" class="cajas" id="cajaNombreProyecto" name="usuario" />
  </div>
  <br />
</div>

<div class="row">
  <div class="col-md-5 col-sm-6 col-xs-12">
    <label style="margin-top: 5px;" class="textoBasico">Plataforma</label><br />
    <select style="width:95%" name="Tipo de Proyecto" class="cajas" id="cajaTipoProyecto">
      <option value=""></option>
    </select>
  </div>
  <div class="col-md-5 col-sm-6 col-xs-12">
```

**Figura 1: Formulario con Bootstrap**

## 2.2.3 jQuery

jQuery [\[9\]](#) es una biblioteca de JavaScript, un fichero con extensión .js que contiene objetos y funciones escritas en JavaScript. Pone a tu disposición muchas funciones prefabricadas para poder ser utilizadas. Por ejemplo, el calendario o el diagrama de tareas del Gestor de Proyectos.

Se ha utilizado jQuery ya que se puede hacer lo mismo que con JavaScript, pero con muchas menos líneas de código, teniendo menos posibilidades de cometer errores.

Por otra parte, JQuery tiene la lógica necesaria para identificar el navegador y la versión de JavaScript permitiendo que funcione en muchas plataformas diferentes. Se ha utilizado esta lógica para mostrar algunos efectos de forma distinta dependiendo de la plataforma.

Una de las principales desventajas de jQuery es actualizar la librería constantemente, puede traer problemas de incompatibilidad con el código, por ello no se han descargado nuevas versiones de la librería.

## 2.2.4 AJAX

“*AJAX* es el acrónimo de Asynchronous Javascript and XML, es decir: *JavaScript y XML Asíncrono*”. [\[10\]](#)

Es una técnica que permite, mediante JavaScript, que un servidor y un navegador intercambien información de forma asíncrona.

Permite que una página web que ya ha sido cargada solicite nueva información al servidor. Cada vez que la página envía información al servidor, recibe de él nueva información sin ser necesario recargar toda la página web.

En el punto 4.2.3 del flujo de datos de la aplicación, se muestra un sencillo ejemplo de una llamada de Ajax.

## 3 Modelo de ciclo de vida, Análisis y Diseño

---

En este capítulo se abordan todos los aspectos relacionados con el modelo de ciclo de vida de la aplicación. El análisis, donde se muestran algunos de los requisitos funcionales y no funcionales, el diagrama de casos de uso y el modelo de datos de la aplicación. Por último, el diseño, donde se muestra el diseño de la arquitectura y de la base de datos.

### 3.1 Modelo de ciclo de vida

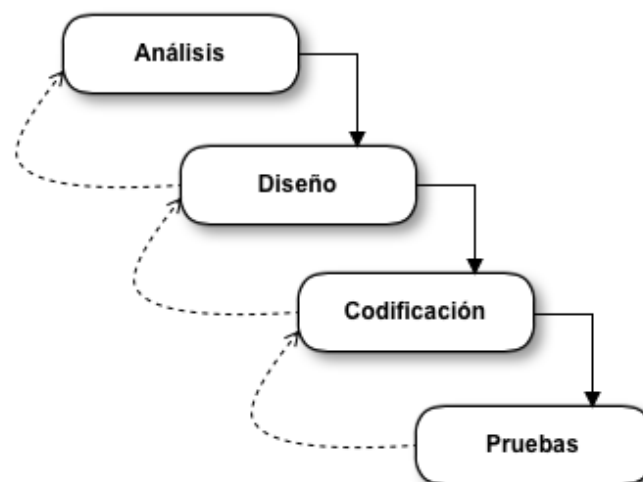
El modelo de ciclo de vida en cascada [11] es un proceso de desarrollo secuencial formado por un conjunto de fases que se ejecutan una tras otra. Se le denomina así debido a que las fases que componen el proyecto se posicionan en forma de cascada, colocadas una encima de otra y siguiendo un flujo de ejecución de arriba hacia abajo.

Este modelo de ciclo de vida será iterativo, pudiendo volver a la etapa anterior cuando sea necesario.

Como desventaja, destacar que un error de diseño nos hace retornar a la fase de dicho error, lo que supone un mayor coste del desarrollo.

Otro inconveniente es que hasta que no se complete una fase no se puede llevar a cabo la siguiente, sin embargo, se ha utilizado este modelo ya que aparte de ser un desarrollo interno para la intranet de la empresa, se consideró que estaban bien marcadas las fases y se conocía a la perfección los recursos y los requisitos solicitados, además de tener más o menos fijado el plazo de finalización del proyecto.

Por estos motivos se descartó el uso de una metodología ágil en la que la comunicación con el equipo y el cliente es primordial, siendo esto inexistente en el desarrollo.



**Figura 2: Modelo de ciclo de vida en cascada**

## 3.2 Análisis

En la parte del Análisis se muestran algunos de los requisitos funcionales y no funcionales de la aplicación, seguidamente se proporciona una definición más formal a través de el Diagrama de Casos de Uso del Sistema y finalmente se muestra el modelo de datos de la aplicación.

### 3.2.1 Especificación de requisitos

En este apartado se detalla el análisis de requisitos realizado. Son presentados en formato estructurado tal y como facilitó la empresa y no en la forma de casos de uso que correspondería a los requisitos en la orientación a objetos.

Los requisitos [\[14\]](#) son aquellas condiciones que debe cumplir la aplicación para conseguir los objetivos planteados en el punto 1 de la memoria.

Existen dos tipos de requisitos, los requisitos funcionales y los no funcionales. Los funcionales son las acciones que el sistema debe cumplir cuando se ejecuta, mientras que los no funcionales son características relacionadas con el funcionamiento.

A continuación, muestro algunos de los requisitos funcionales y no funcionales a modo de ejemplo. Más requisitos funciones en el [Anexo i](#).

#### 3.2.1.1 Requisitos funcionales

RF1	Login del usuario
Descripción	La aplicación permitirá al usuario acceder a la misma con su sesión.
Entrada	El usuario deberá introducir su usuario y contraseña.
Proceso	El sistema comprobará los datos introducidos en la base de datos y si son correctos, el usuario entrará en el sistema.
Salida	Si todo es correcto el sistema redirigirá al usuario a la pantalla de inicio desde la que podrá acceder a sus funciones, en caso contrario el sistema redirigirá al usuario de nuevo a la pantalla de login.

**Tabla 3: Requisito funcional 1**

<b>RF2</b>	<b>Modificar perfil de usuario</b>
<b>Descripción</b>	La aplicación permite al usuario cambiar los datos de su perfil, (nombre, apellidos, fecha nacimiento y contraseña).
<b>Entrada</b>	El usuario en cuestión deberá introducir los campos que desee cambiar.
<b>Proceso</b>	El sistema comprobará los datos introducidos y si es correcto se cambiará la información en la base de datos modificando los campos afectados.
<b>Salida</b>	El sistema informará si el cambio se ha realizado con éxito o informando del error en caso de fallo en los campos introducidos.

**Tabla 4: Requisito funcional 2**

<b>RF3</b>	<b>Filtro de usuarios</b>
<b>Descripción</b>	La aplicación permitirá filtrar usuarios mostrándolos en pantalla.
<b>Entrada</b>	Secuencia de caracteres.
<b>Proceso</b>	El sistema comprobará si existen usuarios en la base de datos con los requisitos del filtro, si los hay, devolverá dichos usuarios.
<b>Salida</b>	Lista de usuarios que cumplen los requisitos del filtro mostrándolos por pantalla con una fotografía del usuario.

**Tabla 5: Requisito funcional 3**

<b>RF4</b>	<b>Dar de alta un proyecto</b>
<b>Descripción</b>	La aplicación permitirá al administrador dar de alta a un proyecto.
<b>Entrada</b>	Los datos del proyecto, como su nombre, tipo, duración estimada, equipo y reuniones.
<b>Proceso</b>	El sistema comprobará los datos introducidos y si son correctos registrará el proyecto en una base de datos.
<b>Salida</b>	El sistema informará si se ha realizado con éxito o con un mensaje de error si ha ocurrido algún error. Si no ha habido errores será añadido a la base de datos.

**Tabla 6: Requisito funcional 4**

<b>RF5</b>	<b>Dar de alta tipo de proyecto</b>
<b>Descripción</b>	La aplicación permitirá al administrador dar de alta tipos de proyectos.
<b>Entrada</b>	El administrador deberá introducir el nombre del tipo de proyecto.
<b>Proceso</b>	El sistema comprobará que no existe ese tipo de proyecto y lo insertará en la base de datos.
<b>Salida</b>	El sistema informará si se ha realizado con éxito o con un mensaje de error si ha ocurrido algún error.

**Tabla 7: Requisito funcional 5**

<b>RF6</b>	<b>Dar de alta un usuario en un proyecto</b>
<b>Descripción</b>	La aplicación permitirá al administrador o al líder de proyecto dar de alta a los usuarios en un proyecto determinado.
<b>Entrada</b>	Nombre de usuario y proyecto.
<b>Proceso</b>	El sistema registrará al usuario como participante del proyecto.
<b>Salida</b>	El sistema informará si se ha realizado con éxito o con un mensaje de error si ha ocurrido algún error.

**Tabla 8: Requisito funcional 6**

<b>RF7</b>	<b>Dar de alta una tarea en un proyecto</b>
<b>Descripción</b>	La aplicación permitirá dar de alta nuevas tareas en un proyecto determinado.
<b>Entrada</b>	El miembro del equipo del proyecto deberá introducir nombre, descripción y duración de la tarea a dar de alta en el proyecto seleccionado.
<b>Proceso</b>	El sistema registrará la tarea como parte del proyecto.
<b>Salida</b>	El sistema informará si la tarea se ha añadido correctamente o error en caso contrario.

**Tabla 9: Requisito funcional 7**



<b>RF8</b>	<b>Actualizar información de un proyecto</b>
<b>Descripción</b>	La aplicación permitirá al administrador cambiar la información de un proyecto.
<b>Entrada</b>	El administrador introducirá los cambios en los datos del proyecto como el nombre, descripción, tipo, equipo o reunión.
<b>Proceso</b>	El sistema comprobará los datos introducidos y si es correcto se cambiará la información del proyecto en la base de datos.
<b>Salida</b>	El sistema informará si el cambio se ha realizado con éxito o informando del error en caso de fallo en los datos introducidos.

**Tabla 10: Requisito funcional 8**

### **3.2.1.2      *Requisitos no funcionales***

A continuación, se muestran los requisitos no funcionales del sistema.

- **(RNF1) Requisitos de Disponibilidad.**

(RNF 1.1) Durante mantenimientos la aplicación no estará disponible.

- **(RNF2) Requisitos de Seguridad.**

(RNF 2.1) La aplicación exigirá contraseñas seguras.

(RNF 2.2) La contraseña se encriptará con un algoritmo de encriptación para evitar robo de identidad (desarrollado por la empresa).

- **(RNF3) Requisitos de Usabilidad.**

(RNF 3.1) La aplicación utilizará tecnologías AJAX para mejorar el tiempo de respuesta.

(RNF 3.2) Si la aplicación falla, se deberá mostrar un mensaje de error informando de lo ocurrido.

- **(RNF4) Requisitos de Backup.**

(RNF 4.1) La aplicación realizará una copia de seguridad de la base de datos para evitar pérdidas de información (realizado por la empresa con todas sus bases de datos)

- **(RNF5) Requisitos de Navegadores.**

(RNF 5.1) La aplicación será compatible con los navegadores web Chrome, Firefox, Safari e Internet Explorer.

- **(RNF6) Diseño responsive.**

(RNF 6.1) La aplicación debe tener un diseño responsive para conseguir una correcta visualización en varios tipos de dispositivos.

- **(RNF7) Autorización de usuarios.**

(RNF 6.1) La aplicación mostrará u ocultará opciones dependiendo del tipo de usuario que se haya autenticado en el sistema.

### **3.2.2 Diagrama de casos de Uso**

Un diagrama de caso de uso describe las actividades que se deben realizar para llevar a cabo algún proceso. Los personajes o entidades que realizan dichas acciones en un diagrama de caso de uso se denominan actores. En la aplicación existen tres actores: el miembro de un equipo de proyecto, el líder del equipo de un proyecto y el administrador.

Los casos de uso son las actividades o acciones desarrolladas por los actores.

Los colaboradores externos desempeñan las mismas acciones que los miembros de equipo con la única particularidad de que son trabajadores externos a la empresa, contratados para desarrollar algunas tareas específicas.

A continuación, se muestra el Diagrama de Casos de Uso de la aplicación Gestor de Proyectos:

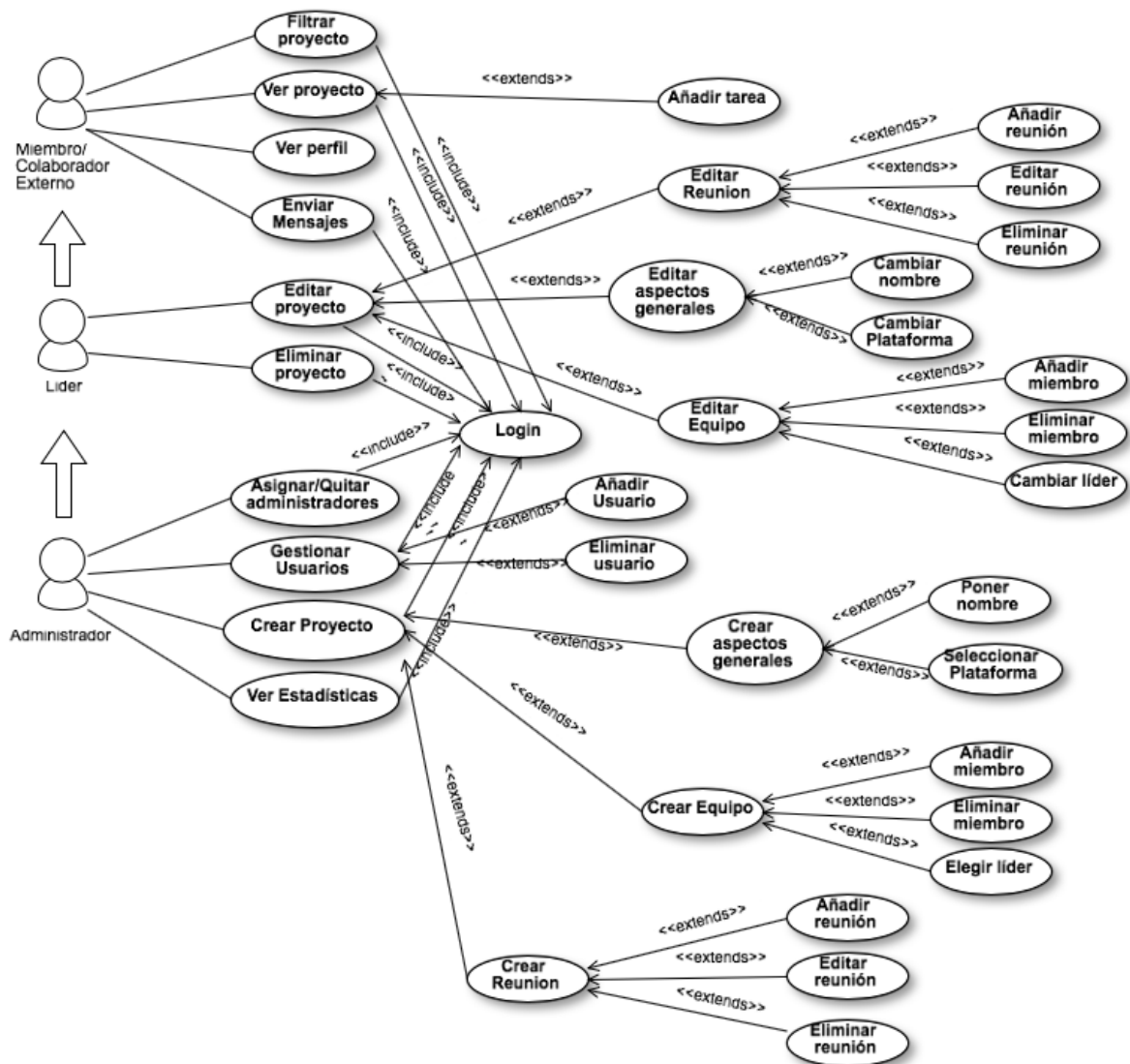
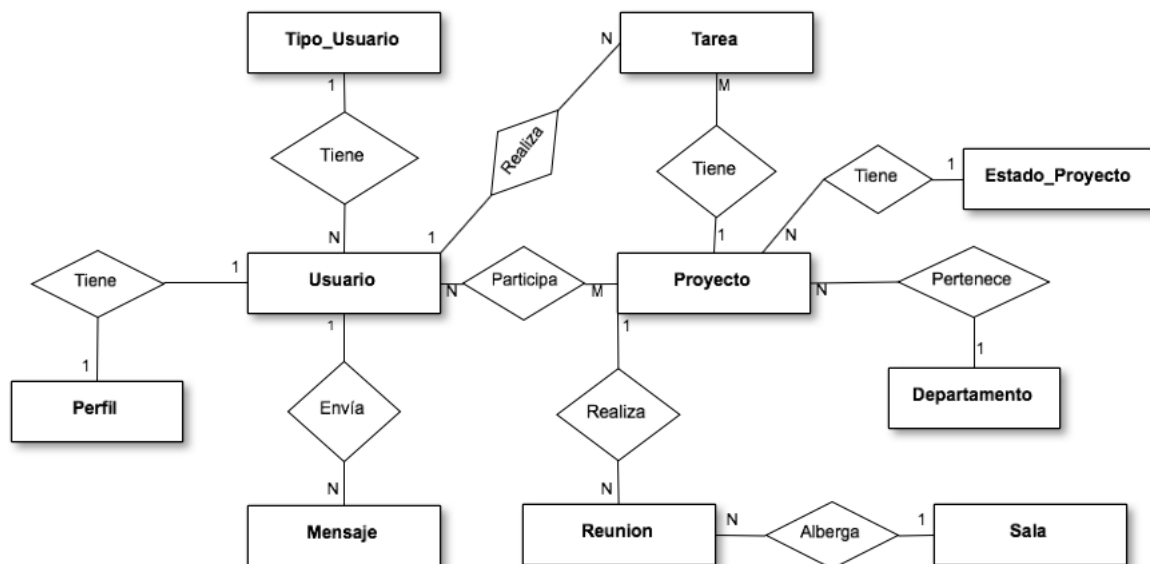


Figura 3: Diagrama de casos de uso

### 3.2.3 Modelo de datos

En el Gestor de Proyectos y en la empresa utilizamos una base de datos relacional ya que, para el uso que se le quiere dar, una base de datos no relacional no presentaría ninguna ventaja. A continuación, se muestra el modelo entidad-relación de la aplicación (para no poner los atributos en el modelo de datos, estos se muestran en el diseño de la base de datos).



**Figura 4: Modelo Entidad-Relación**

A continuación, se describen las entidades del modelo entidad-relación:

- **Usuario:** Tabla que representa a los usuarios de la aplicación. Cada usuario tiene un tipo de usuario y un perfil propio. Estará relacionada con los proyectos y tareas en las que participa.
- **Perfil:** Tabla que representa el perfil del usuario.
- **Tipo\_Usuario:** Tabla que representa los tipos de usuarios de la aplicación en función del trabajo desempeñado: Administrador, Líder del proyecto, Miembro y Colaborador externo.
- **Proyecto:** Tabla con los proyectos. Estos proyectos se relacionan con usuarios, tareas y el departamento al que pertenece.
- **Tarea:** Tabla que contiene las tareas de la aplicación. Se relacionan con el proyecto al que pertenece y el usuario que la realiza.
- **Departamento:** Tabla con los departamentos sobre los que actuará cada proyecto. Relacionada con proyecto.
- **Reunión:** Tabla con las reuniones de un proyecto. Están relacionadas con el proyecto al que pertenece y la sala donde se realiza.
- **Mensaje:** Tabla que contiene los mensajes. Se relaciona con Usuario para saber el origen y destino de dicho mensaje.
- **Estado\_Proyecto:** Tabla que contiene los diferentes estados en los que se puede encontrar un proyecto. Se relaciona con proyecto.
- **Sala:** Tabla que contiene las salas donde se realizan las reuniones.

### 3.3 Diseño

En este último apartado se presenta la información del diseño del sistema desarrollado. Inicialmente se proporciona una visión arquitectónica y a continuación se detalla la arquitectura de la aplicación software mediante un diagrama con el diseño de las tablas.

### 3.3.1 Diseño de la arquitectura

El objetivo principal de la aplicación es facilitar la gestión de proyectos repartiendo las tareas entre los integrantes del equipo. Además, es posible obtener informes y gráficas con estadísticas simples acerca de cada proyecto o usuario en concreto.

- En lo relacionado con el usuario, destacar que se permite el registro del mismo, el posterior login, su filtrado y por último, la visualización y edición del perfil y los distintos proyectos en los que participa.
- En lo relacionado con los proyectos, destacar que se permite la visualización, creación y edición de un proyecto con su nombre, tipo y duración estimada, equipo que lo compone con su líder de proyecto, diagrama con las tareas realizadas por los miembros del equipo, calendario con reuniones del proyecto y un filtro de proyectos por distintos campos.
- En lo relacionado con las estadísticas e informes, se podrá ver el estado de los proyectos con el porcentaje de desarrollo de estos, se podrá ver el desfase entre el tiempo estimado y tiempo real y se podrá ver el trabajo realizado por el usuario en el día, pudiendo ver cuánto tiempo a dedicado a cada proyecto y sus tareas realizadas.

Para el desarrollo del proyecto se ha utilizado ASP.NET MVC, framework que utiliza el patrón Modelo-Vista-Controlador, es decir, se deben seguir las reglas impuestas por este patrón.

En el Visual Studio tendremos tres carpetas:

- **Modelo:** en él se implementan las clases y estructuras de datos que resuelven la lógica de nuestra aplicación. La mayoría de la lógica de la aplicación se resuelve en el modelo mediante procedimientos almacenados que cuelgan de la carpeta Programación/Procedimientos Almacenados.
- **Vista:** se define cómo se mostrará de forma visual la interfaz de usuario de la aplicación. En la aplicación las vistas principales son: vistaPrincipal, verProyecto, crearProyecto y editarProyecto con sus respectivas llamadas de AJAX para solicitar información al servidor.
- **Controlador:** es un conjunto de clases en donde se coordina la comunicación entre las peticiones que hace el cliente, el modelo que procesa los datos que llegan del cliente y comunica a las vistas para que muestren los datos solicitados por el cliente. En la aplicación los controladores principales son HomeController, vistaPrincipalController, verProyectoController, crearProyectoController, editarProyectoController y estadisticasController con sus respectivos métodos que ejecutan el procedimiento almacenado en base de datos que corresponda según la vista en la que nos encontremos.

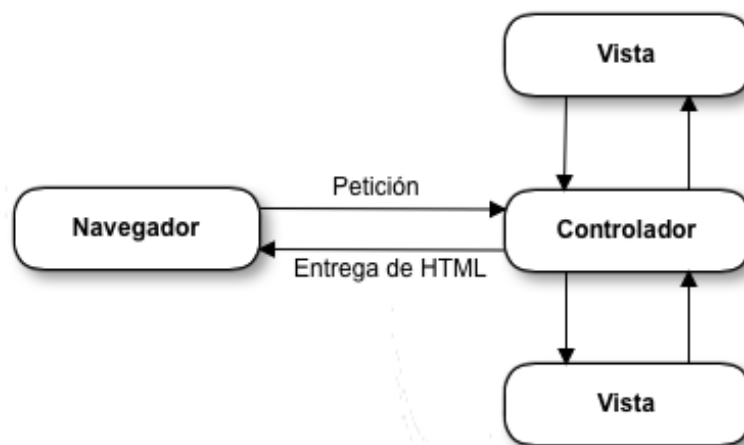


Figura 5: M-V-C

### 3.3.2 Diseño de las tablas

A continuación, se muestra un ejemplo con el diseño de las tablas de la base de datos.



Figura 6: Diseño Base de Datos

## 4 Desarrollo

---

En este capítulo se exponen los detalles relativos a la fase de desarrollo e implementación de la aplicación.

Tanto desde casa, como en la oficina de la empresa, no se trabaja en local.

A través de internet, se utiliza la máquina virtual **VMware horizon client** con usuario, contraseña y dominio para trabajar en la nube, accediendo a una red privada en la que, dependiendo del usuario de acceso, entras con tu perfil. De esta forma, podremos trabajar en el servidor como si trabajásemos en local.

Se trabaja sobre una ip destinada a desarrollar el código y probarlo hasta que funcione correctamente y pase las pruebas antes de entrar a producción. También se utiliza una ip para acceder a los informes usados por el gestor de Proyectos.

Como se ha comentado anteriormente, la aplicación se realiza usando el framework .net, utilizando PostgreSQL como gestor de base de datos usando el lenguaje TSQL. Se utiliza el entorno de desarrollo Microsoft Visual Studio 2017 usando c#, JavaScript, HTML y CSS. Son los entornos y lenguajes usados en la empresa y por ello han sido utilizados, pudiendo haber optado por otros métodos. Sin embargo, me ha parecido una buena elección y una forma rápida y sencilla de realizar el proyecto.

### 4.1 Hardware

La implementación de la aplicación se ha desarrollado en un Intel i7 de sexta generación con 16gb de memoria RAM. También se ha trabajado desde casa con un ordenador portátil MacBook Pro de 2,7 GHz Intel Core i7 con memoria de 4GB.

No ha sido necesario el uso de ningún otro hardware especial salvo una Tablet para comprobar el diseño en ella.

### 4.2 Software

Se ha empleado una gran variedad de herramientas software para el desarrollo de la aplicación.

Como el sistema operativo es de MAC OS se ha utilizado VMware Horizon Client tanto para trabajar en la empresa como trabajando desde casa.

**VMware Horizon Client** [\[13\]](#) es un software que establece una conexión entre los dispositivos de punto final y los escritorios y aplicaciones virtuales de Horizon View.

Esta aplicación se comunica con View Connection Server, que actúa como un intermediario entre el dispositivo cliente y los escritorios de View. Los usuarios ingresan credenciales en Horizon Client y View Connection Server, los autentifica y luego encuentra sus escritorios virtuales.

**Virtualización de escritorio** [\[15\]](#) describe el proceso de separación entre el escritorio con los datos y programas que utilizan los usuarios para trabajar, de la máquina física. El escritorio "virtualizado" es almacenado remotamente en un servidor central en lugar de en el disco duro del ordenador personal, permitiendo a los usuarios acceder y trabajar remotamente a sus escritorios desde cualquier dispositivo capaz de conectarse remotamente al escritorio.

**Windows 10 2016:** Sistema operativo de Microsoft sobre el que se han ejecutado todos los entornos de desarrollo y las herramientas comentadas. Se ha utilizado Windows porque es el sistema operativo empleado en la empresa para poder usar el Visual Studio y el SQL Server.

#### 4.2.1 Entornos de desarrollo

“**Microsoft Visual Studio** es un entorno de desarrollo integrado (IDE) para sistemas operativos Windows. Soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como **ASP.NET MVC**, Django, etc.”

“Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET. Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y consolas, entre otros”. [\[17\]](#)

“**Microsoft SQL Server Management Studio 17** es un sistema de manejo de bases de datos del modelo relacional, desarrollado por la empresa Microsoft. El lenguaje de desarrollo utilizado es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL).” [\[18\]](#)

Permite el soporte de transacciones, el soporte de procedimientos almacenados y trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.

“**Business Intelligence Development Studio** es el antiguo IDE de Microsoft y se usa para desarrollar soluciones de análisis de datos y Business Intelligence utilizando los servicios de análisis de Microsoft SQL Server, Reporting Services y Integration Services. Se basa en el entorno de desarrollo de Microsoft Visual Studio, pero se personaliza con las extensiones y tipos de proyectos específicos de los servicios de SQL Server.” [\[19\]](#)

#### 4.2.2 Lenguajes de programación

“**HTML**, HyperText Markup Language, emplea el lenguaje marcado para el desarrollo de páginas web” [\[20\]](#). En el archivo HTML del componente, se definirá la vista del componente. En el proyecto desarrollado, el código HTML se combinará con el código del framework Bootstrap.



Para añadir un elemento externo a la página, este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. Se ha utilizado este lenguaje para diseñar la vista de la aplicación.

“CSS son las siglas de Cascading Style Sheets que es un lenguaje de diseño gráfico que emplea un lenguaje marcado para el diseño de los elementos visuales declarados en las vistas Html, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra..., modificando la apariencia de una página web de una forma más sencilla, permitiendo a los desarrolladores controlar el estilo y formato de sus documentos.” [\[21\]](#)

Cada regla consta de un selector y una declaración, esta última va entre corchetes y consiste en una propiedad o atributo y un valor separados por dos puntos. Se ha utilizado este lenguaje para dar estilo a la vista.

```
.gantt_grid_scale div:nth-child(4) {  
    display: none;  
}  
  
.gantt_row > div:nth-child(4) {  
    display: none;  
}  
  
.gantt_grid_scale div:nth-child(3) {  
    display: none;  
}
```

**Figura 7: Ejemplo CSS**

“LESS es un lenguaje dinámico de código abierto de hojas de estilo que puede ser compilado en Hojas de estilo en cascada (CSS) y ejecutarse en el lado del cliente o en el lado del servidor.”

“La principal diferencia entre Less y otros precompiladores CSS es que Less permite la compilación en tiempo real vía less.js por el navegador. LESS se puede ejecutar en el lado del cliente y del lado del servidor o se puede compilar en CSS sin formato.” [\[22\]](#)

LESS permite que se definan las variables. Las variables de LESS se definen con una arroba “@” y la asignación se hace con dos puntos “:”. Ejemplo:

```
@color: #003366;  
@background: #d9d9d9;  
  
#header {  
    color: @color;  
    background: @background;  
}  
  
h2 {  
    color: @color;  
}
```

El código anterior en LESS compilaría en el siguiente código CSS:

```
#header {  
  color: #003366;  
  background: #d9d9d9;  
}  
h2 {  
  color: #003366;}
```

LESS permite operaciones y funciones. Las operaciones permiten la suma, resta, división y multiplicación de valores de propiedades y colores, que se pueden utilizar para crear relaciones complejas entre propiedades.

CSS admite anidamiento lógico, pero los bloques de código no están anidados. LESS permite anidar los selectores dentro de otros selectores, esto hace la herencia clara y las hojas de estilo más cortas.

**“Lenguaje C#** es un lenguaje de programación orientado a objetos, desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.” [\[25\]](#)

El lenguaje C# forma parte de la plataforma .NET, pero este es una **API**, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Se ha utilizado este lenguaje para los controladores de la aplicación.

**“Transact-SQL** es el principal medio de interacción con el Servidor, permite realizar las operaciones claves en SQL Server, incluyendo la creación y modificación de esquemas de base de datos, inserción y modificación de datos en la base de datos, así como la administración del servidor como tal.” [\[23\]](#)

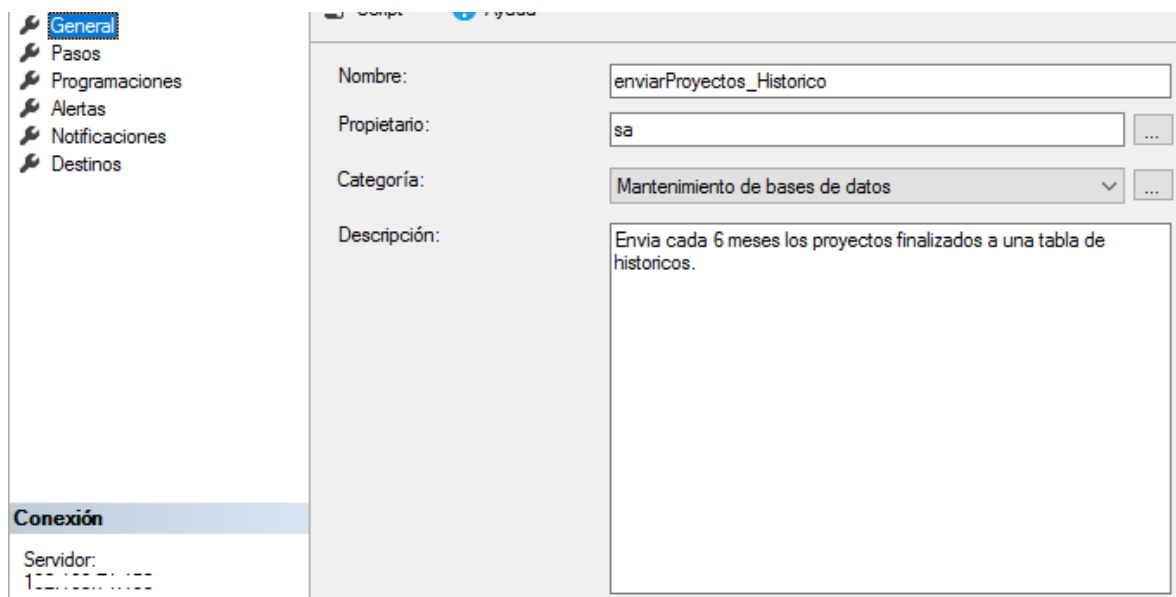
Estas operaciones se realizan mediante el envío de sentencias en T-SQL y declaraciones procesadas por el servidor y los resultados o errores regresan a la aplicación cliente. Se ha utilizado este lenguaje para el modelo de la aplicación.

**“Las funciones** son un objeto que combina algunas capacidades de las vistas con otras de los procedimientos. Pueden extraer datos y ejecutar cálculos y devuelven un resultado al usuario o al programa que les ejecutó”. [\[23\]](#)

Como los procedimientos, incluyen códigos de TSQL y pueden ser ejecutados con parámetros. Las funciones devuelven un valor o un conjunto de valores. En el Gestor de proyectos se han utilizado funciones que convierten fechas y horas al formato deseado.

**Un trabajo (job)** [\[23\]](#) o tarea de SQL Server, es un proceso automático que se ejecuta automáticamente en una programación establecida bajo control del Agente SQLServer. Mediante el uso del Agente SQL Server para programar, se puede determinar el orden en que se realizarán las tareas administrativas.

En la parte de general, pondremos el nombre y la descripción del procedimiento. En pasos, se lanza el procedimiento almacenado a ejecutar. Por último, en programación pondremos cada cuanto sucede el trabajo.



**Figura 8: Ejemplo Trabajo**

“Un **procedimiento almacenado de SQL Server** es un grupo de una o varias instrucciones Transact-SQL o una referencia a un método de Common Runtime Language (CLR) de Microsoft.NET Framework.” [24] Los procedimientos pueden:

- Aceptar parámetros de entrada y devolver varios valores en forma de parámetros de salida al programa que realiza la llamada.
- Contener instrucciones de programación que realicen operaciones en la base de datos. Pueden contener llamadas a otros procedimientos.

“**Lenguaje Integrated Query (LINQ)** es un componente de la plataforma Microsoft .NET que agrega capacidades de consulta a datos de manera nativa a los lenguajes .NET con sintaxis parecida a SQL.” [26]

En LINQ es necesario crear las clases, debido a que si, por ejemplo, quiero traerme un equipo de un proyecto, debo de instanciar un objeto de tipo equipo para hacer las operaciones que necesite sobre el mismo.

### **Procedimientos almacenados vs LINQ [12]**

La mayoría del trabajo ha sido realizado mediante procedimientos almacenados debido a que, además de ser la forma más común de trabajar en la empresa, cabe destacar, que a pesar de que no habrá diferencia de rendimiento con consultas simples, presenta algunas ventajas con respecto a LINQ, aunque también alguna pequeña desventaja.

Entre las **ventajas**, cabe destacar:

- Un tráfico de red reducido entre el cliente y el servidor.
- Mayor seguridad.
- Fácil reutilización del código (encapsulación de procedimientos).
- Rendimiento mejorado debido a que un procedimiento se compila la primera vez que se ejecuta y crea un plan de ejecución que vuelve a usarse en posteriores ejecuciones y necesita menos tiempo para procesar el procedimiento.

- Por último, la ventaja más importante es que cuando las aplicaciones cliente llaman a procedimientos y mantienen las operaciones de base de datos en la capa de datos, solo deben actualizarse los cambios de los procesos en la base de datos subyacente. El nivel de aplicación permanece independiente y no tiene que tener conocimiento sobre los cambios realizados en los diseños, las relaciones o los procesos de la base de datos.

Entre las **desventajas**, cabe destacar:

- Realizando la lógica a nivel de aplicación, se evita resolver muchos problemas con TQSL. En alguna parte del desarrollo me he quedado atascado buscando información de cómo hacerlo en el procedimiento almacenado a pesar de saberlo hacer a nivel de aplicación.
- Por último, es más fácil depurar con el Visual Studio ya que en los procedimientos es necesario dar valores a los parámetros de entrada del mismo para ir evaluando el comportamiento.

### 4.2.3 Flujo de datos

En este apartado se muestra un ejemplo del flujo de datos en la aplicación.

En nuestro ejemplo, se muestra el camino seguido desde que solicitas quién es la persona responsable de una tarea específica hasta que recibes la respuesta.

En la vista tenemos las llamadas de AJAX que realizan una petición al controlador.

En los controladores tendremos las llamadas a los procedimientos almacenados de la base de datos.

```
jQuery.ajax({
  async: false,
  //url: '/verProyecto/dameUsuarioTarea',
  url: '@Url.Action("dameUsuarioTarea", "verProyecto")',
  type: 'POST',
  data: { 'id': (Value) },
  datatype: "json",
  success: function (response) {
    $('#usuarioTarea2').val(response);
  }
});
```

**Figura 9: Llamada Ajax**

En la figura 10 se realiza una llamada desde el controlador vistaController, el cual manda los datos a otro controlador en el que están todas las llamadas a la base de datos.

Este segundo controlador llamado Gestor\_SP, se utiliza para mantener más limpio el controlador principal, limpiando los datos que le llegan y realizando algunas funciones sobre

ellos para evitar trabajo y líneas de código al primer controlador. Con esta forma de hacerlo, el flujo se adapta a la forma de trabajar de la empresa, siguiendo así el esquema del resto de desarrollos de la misma.

```
public string dameUsuarioTarea(int id)
{
    string nombreUsuarioTarea = Gestor_SP.verProyecto.dameUsuarioTarea(id);

    return nombreUsuarioTarea;
}
```

**Figura 10: Llamada desde controlador al controlador Gestor\_SP**

En la figura 11 se realiza una llamada al procedimiento almacenado SP\_dameUsuarioTarea desde el controlador Gestor\_SP.

```
public static string dameUsuarioTarea (int id)
{
    return INT_SP.procedimientos_almacenadosSP.ExecuteCadena("SP_dameUsuarioTarea",
        new SqlParameter("@ID", id)
    );
}
```

**Figura 11: Llamada al procedimiento almacenado**

En la figura 12 se muestra la función ExecuteCadena que hace la conexión con la base de datos e indica que el valor devuelto será una cadena.

```

public static string ExecuteCadena(string storedProcedureName, params SqlParameter[] arrParam)
{
    string cadena = "";
    // Open the connection
    using (SqlConnection cnn = new SqlConnection(System.Configuration.ConfigurationManager.ConnectionStrings["GestorProyectos"].ConnectionString))
    {
        cnn.Open();

        // Define the command
        using (SqlCommand cmd = new SqlCommand())
        {
            cmd.Connection = cnn;
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.CommandText = storedProcedureName;
            cmd.CommandTimeout = 0;
            // Handle the parameters
            if (arrParam != null)
            {
                foreach (SqlParameter param in arrParam)
                {
                    cmd.Parameters.Add(param);
                }
            }

            // Define the data adapter and fill the dataset
            using (SqlDataAdapter da = new SqlDataAdapter(cmd))
            {
                cadena = da.SelectCommand.ExecuteScalar().ToString();
            }
        }
    }
    return cadena;
}

```

**Figura 12: Función executeCadena**

Por último, se muestra el procedimiento almacenado que devuelve el usuario solicitado a los controladores de tal forma que finalmente pueda ser tratado en el response del AJAX.

```

USE [GestorProyectos]
GO
/***** Object: StoredProcedure [dbo].[SP_dameUsuarioTarea]    Script Date: 04/03/2019 16:59:29 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Miguel>
-- Create date: <28/11/2018>
-- Description: <login>
-- =====
ALTER PROCEDURE [dbo].[SP_dameUsuarioTarea]
    @ID int
AS
BEGIN
    SET NOCOUNT ON;

    select Usuario from Tarea where ID = @ID
END

```

**Figura 13: Procedimiento almacenado**

#### 4.2.4 Uso de código externo: Plugins

Se ha utilizado código externo para facilitar la implementación en ciertas partes del código. Un Plugin es una aplicación o programa informático que se relaciona con otro para agregarle una función nueva y generalmente muy específica.

En la aplicación se han utilizado diversos plugins, entre ellos cabe destacar:

- Fullcalendar.js: para mostrar el calendario de reuniones.
- datetimepicker.js: para introducir fechas y horas en los lugares donde se requiere.
- jsGantt.js: para mostrar el diagrama de las tareas de cada proyecto.
- progressBar.js: para mostrar la barra de progreso del proyecto.

El plugin datetimepicker para insertar una fecha y una hora:

```
<script src= 'build/jquery/jquery.datetimepicker.full.min.js'></script>
```

Pintar el plugin en html:

```
<input id = "datetimepicker" type = "text" >
```



**Figura 14: Insertar fecha**

Funciones utilizadas para el desarrollo del calendario de reuniones y la barra de progreso del proyecto.

```
events: function (start, end, callback) {
    $.ajax({
        type: "POST",
        url: "webserv.aspx/TestOnWebService",
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        success: function (doc) {
            alert("Success");
            var events = [];
            alert(doc.d);
            var obj = $.parseJSON(doc.d);
            console.log(obj);
            $(obj.event).each(function () {
                events.push({
                    title: $(this).attr('title'),
                    start: $(this).attr('start'),
                    end: $(this).attr('end'),
                    id: $(this).attr('id')
                });
            });
            //callback(events);
            callback && callback(events);
        },
        error: function(xhr, status, error) {
            alert(xhr.responseText);
        }
    });
}
```

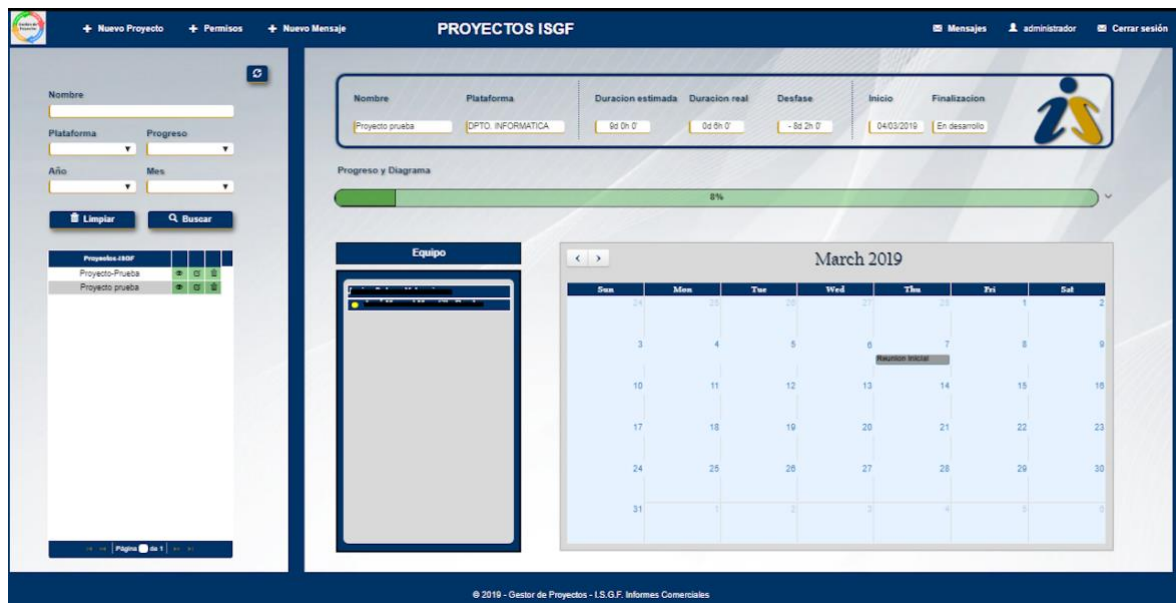
**Figura 15: Función Calendario**

```
<script>
    $(document).ready(function () {
        $(function () {
            $("#progressbar").progressbar({
                value: 10
            });
        });
    });
</script>

<style>
    #progressbar {
        height: 50px;
        width: 100%;
    }
</style>
<div id="progressbar"></div>
```

**Figura 16: Función Barra Progreso**

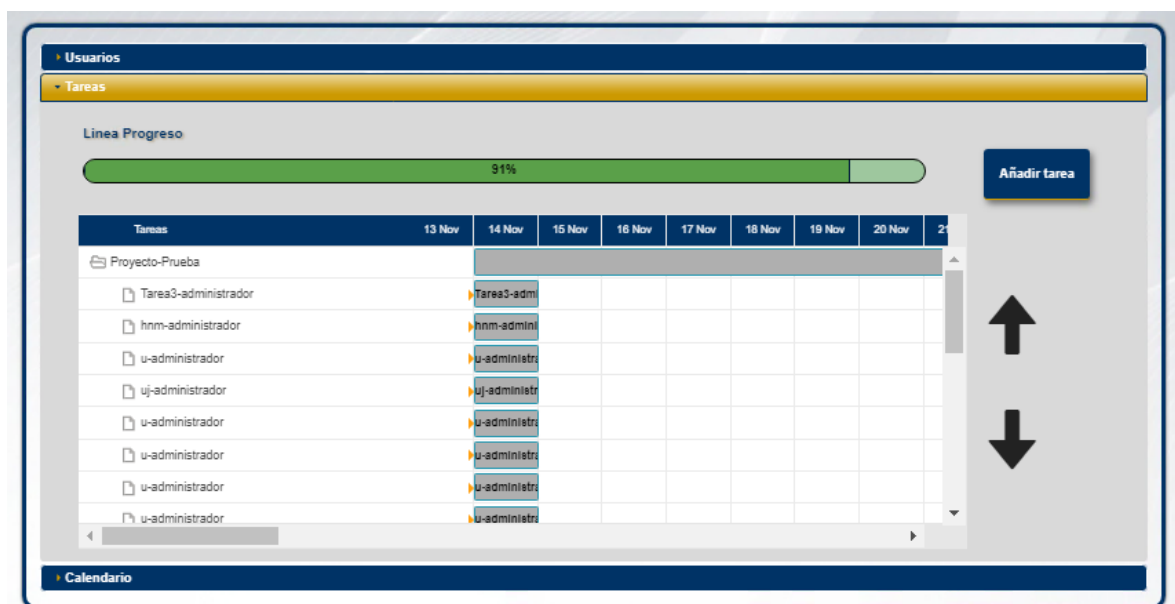
A continuación, se muestra el calendario y la barra de progreso del Proyecto Prueba.



**Figura 17: Vista proyecto**

Además, en la figura 17 podemos ver los campos del Proyecto prueba y el equipo que lo conforma, entre ellos el campo desfase, el cual nos irá indicando si el proyecto termina antes de lo esperado o si está sufriendo un retraso.

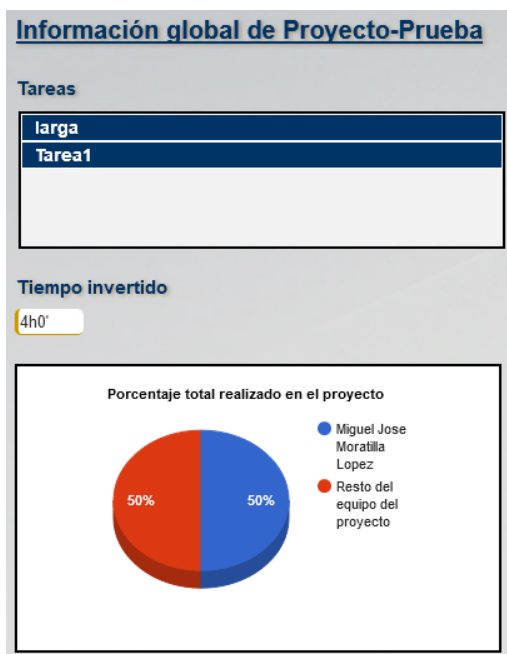
A continuación, se muestra el diagrama de tareas del Proyecto prueba. En dicho diagrama podremos acceder a la información de cada tarea, pudiendo ver el empleado, duración y día de la misma.



**Figura 18: Diagrama de tareas del proyecto**

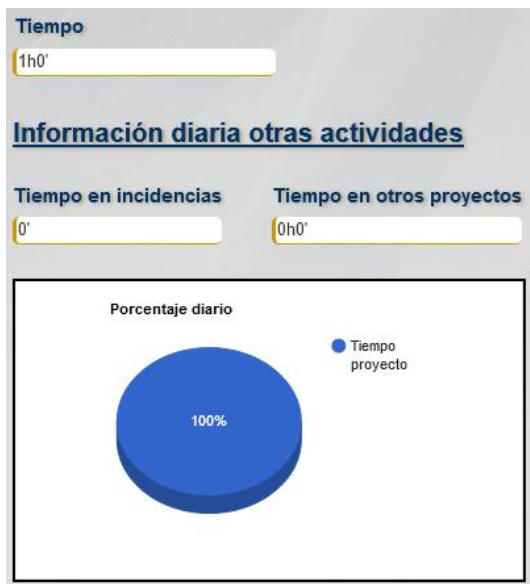
El último uso de plugins es para realizar las gráficas con estadísticas.





La primera muestra las tareas realizadas por un miembro en un proyecto específico y el tiempo invertido en él.

**Figura 19: Gráfica Proyecto-Usuario**



La segunda muestra el tiempo empleado en el desarrollo del proyecto en el día seleccionado en proporción al tiempo empleado en otros proyectos distintos e incidencias del día. (El tiempo de resolución diaria de incidencias se lee de otra aplicación de la empresa que no tiene que ver con el desarrollo de este proyecto).

**Figura 20: Gráfica Diaria-Usuario**

#### 4.2.5 Conceptos de desarrollo

**Uso de ViewBag:** es un objeto de tipo dinámico que sirve como transporte de datos entre el controlador y la vista.

El modelo MVC puede pasar solo un objeto a la vista, pero a veces la vista necesita más información y por ello se usa ViewBag, así, se puede pasar cualquier objeto o dato adicional que se necesite en la vista, por ejemplo, un booleano para controlar si la vista se abre de un modo u otro.

**Uso del document ready:** se ejecuta cuando el contenido HTML está listo, antes de las imágenes y otros recursos. A diferencia de window onload, que se ejecuta al final del proceso de carga del documento. Es decir, cuando todos los objetos han terminado de cargarse (las hojas de estilo no siempre están cargadas al momento de lanzarse este evento).

**Uso de html dinámico** en JavaScript. Por ejemplo, al pintar los miembros de un equipo o al filtrar usuarios en la aplicación.

```
jQuery.ajax({
  async: false,
  //url: '/verProyecto/filtrarUsuarios',
  url: '@Url.Action("filtrarUsuarios", "verProyecto")',
  type: 'POST',
  data: { 'cajaNombreUsuario': $('#cajaNombreUsuario').val() },
  success: function (response) {
    for (var i = 0; i < response.length; i++) {
      $("#sortable2").append("<li id='" + response[i].Value + "' class= 'ui-state-default'" + response[i].Text + "</li>");
    }
  }
});
```

**Figura 21: Ejemplo html dinámico para filtrar usuarios**

**Uso de Grid:** es un mecanismo utilizado para la representación de datos en tablas compuestas de filas y columnas. Bootstrap permite hasta 12 columnas por página. Además, si no necesitas las 12 columnas individualmente, las columnas se pueden agrupar para crear columnas más anchas.

El sistema de cuadrícula de Bootstrap es responsive y las columnas se reorganizarán según el tamaño de la pantalla, en la pantalla grande del lugar de trabajo puede verse mejor con el contenido organizado en tres columnas, pero en una pantalla pequeña como la de la Tablet sería mejor si los elementos de contenido se apilaran encima del otro. Por ello no fue necesario hacer prácticamente cambios en la versión para Tablet.

```

$("#grid").jqGrid({
  postData: {
    'nombreProyecto': function () { return jQuery('#cajaNombreProyecto').val() },
    'IDtipoProyecto': function () { return jQuery('#cajaTipoProyecto').val() },
    'IDAñoFechaProyecto': function () { return jQuery('#cajaAñoFechaProy').val() },
    'IDMesFechaProyecto': function () { return jQuery('#cajaMesFechaProy').val() },
    'IDbitTerminado': function () { return jQuery('#bitTerminado').val() }
  },
  //url: '/proyectos/filtrarNombreProyecto',
  url: '@Url.Action("filtrarNombreProyecto", "proyectos")',
  datatype: 'json',
  mtype: 'POST',
  gridview: true,
  autowidth: true,
  sortable: true,
  resizable: false,
  colNames: ['Proyectos-ISGF', '', '', '', ''],
  colModel: [
    { name: 'Nombre', align: "center", width: "225" },
    { name: 'IDtipoProyecto', align: "center", width: "0", hidden: true },
    { name: 'verProyecto', align: "center", width: "30" },
    { name: 'editar', align: "center", width: "30" },
    { name: 'papelera', align: "center", width: "30" }
  ],
  rowNum: 50,
  pager: $('#pager'),
  viewrecords: true,
  height: '445',
  width: '315',
  caption: 'PROYECTOS-ISGF',
  gridComplete: function () {
    $('#grid tbody tr td:nth-child(3)').css("background-color", "rgb(41, 163, 41, 0.5)");
  }
});

```

**Figura 22: Ejemplo Grid del filtro de proyectos**

**Uso de Layout:** plantilla a partir de la cual la página web comienza a desarrollar sus contenidos específicos. Contiene las referencias a los .css y .js.

Ejemplo referencias css:

```

<link href="~/Content/bootstrap-3.3.7-dist/css/bootstrap.css" rel="stylesheet" />
<link href="@Url.Content("~/Content/jquery-ui.css")" rel="stylesheet" type="text/css" />

```

Ejemplo referencias js:

```

<script src="~/Scripts/jquery-1.10.2.js"></script>
<script src="~/fullcalendar-3.3.1/lib/moment.min.js"></script>
<script src="~/fullcalendar-3.3.1/fullcalendar.min.js"></script>

```

Ejemplo **footer** de la página usando layout:

```

<footer style="clear:both">
  <p>&copy; @DateTime.Now.Year - Gestor de Proyectos - I.S.G.F. Informes Comerciales</p>
</footer>

```

**RenderBody:** se utiliza en layout.cshtml para indicar que en ese punto se incluirá el código generado por la vista. Todos los layouts deben incluirla.

```
<div style="height:30%; width:100%; ">  
    @RenderBody()  
</div>
```

**RenderSection:** incluimos en nuestra plantilla secciones o bloques de código Razor / HTML que se han definido en la vista. Además, se puede indicar si es obligatorio o no definirla. Por defecto el valor se encuentra a true, por lo que, si no la incluimos en nuestra vista, dará un mensaje de error.

```
@*@RenderSection("scripts", required: false)*@
```

Ficheros de configuración en el [Anexo iii](#) y Organización de las carpetas en el [Anexo iv](#).

## 5 Integración, pruebas y resultados

A continuación, se describen las pruebas realizadas sobre la aplicación para asegurar su correcto funcionamiento. Definición del tipo de prueba extraído de [\[16\]](#).

### 5.1 Pruebas Unitarias

Este tipo de pruebas comprueban la lógica, la funcionalidad y si se cumplen las especificaciones para cada componente de la aplicación.

Para cada módulo de la aplicación se ha realizado una prueba unitaria para asegurar que los fallos no son producidos por módulos ajenos al mismo y así tener localizados estos fallos. Se ha comprobado el comportamiento con el paso de distintos argumentos y se han analizado los valores devueltos.

### 5.2 Pruebas Funcionales

Son las pruebas que aseguran el cumplimiento de los requisitos funcionales de la aplicación expuestos en el apartado del diseño. Más pruebas en el [Anexo ii](#).

Prueba RF1	Login del usuario
Descripción	Esta prueba se encarga de comprobar que el login por parte de un usuario funciona correctamente.
Entrada	El usuario deberá introducir su usuario y contraseña.
Resultado Esperado	El usuario hace login correctamente y se crea una variable de sesión en el navegador.
Prueba	Correcta.

Tabla 11: Prueba RF1

Prueba RF2	Modificar perfil de usuario
Descripción	Prueba que permite al usuario cambiar los datos de su perfil, (nombre, apellidos, fecha nacimiento y contraseña).
Entrada	El usuario en cuestión deberá introducir los cambios en los campos correspondientes.
Resultado Esperado	El perfil del usuario se actualiza con los nuevos datos.
Prueba	Correcta.

Tabla 12: Prueba RF2

Prueba RF3	Filtro de usuarios
Descripción	Esta prueba se encarga de comprobar que el filtro de usuarios se realiza correctamente.
Entrada	Secuencia de caracteres.
Resultado Esperado	La aplicación muestra un listado de usuarios filtrados correctamente con esos caracteres.
Prueba	Correcta.

**Tabla 13: Prueba RF3**

Prueba RF4	Dar de alta un proyecto
Descripción	Esta prueba se encarga de comprobar que el administrador da de alta nuevos proyectos.
Entrada	Los datos del proyecto, como su nombre, tipo, duración estimada, equipo y reuniones.
Resultado Esperado	El proyecto se crea correctamente con toda la información en la base de datos.
Prueba	Correcta.

**Tabla 14: Prueba RF4**

Prueba RF5	Dar de alta tipo de proyecto
Descripción	Prueba que permite al administrador dar de alta tipos de proyectos.
Entrada	El administrador deberá introducir el nombre del tipo de proyecto.
Resultado Esperado	El tipo de proyecto es creado en la base de datos.
Prueba	Correcta.

**Tabla 15: Prueba RF5**

Prueba RF6	Dar de alta un usuario en un proyecto
Descripción	Esta prueba se encarga de comprobar que el administrador o líder de proyecto puede dar de alta a un usuario en un equipo del proyecto.
Entrada	Nombre de usuario y proyecto.
Resultado Esperado	El usuario se añade al equipo correctamente.
Prueba	Correcta.

**Tabla 16: Prueba RF6**

<b>Prueba RF7</b>	<b>Dar de alta una tarea en un proyecto</b>
<b>Descripción</b>	Prueba que permite dar de alta nuevas tareas en un proyecto concreto.
<b>Entrada</b>	El usuario del proyecto deberá introducir nombre, descripción y duración de la tarea a dar de alta en el proyecto seleccionado.
<b>Resultado Esperado</b>	La tarea es creada con toda la información en la base de datos.
<b>Prueba</b>	Correcta.

**Tabla 17: Prueba RF7**

<b>Prueba RF8</b>	<b>Actualizar información de un proyecto</b>
<b>Descripción</b>	Prueba que permite al administrador o líder de proyecto cambiar la información de un proyecto.
<b>Entrada</b>	El administrador introducirá los cambios en los datos del proyecto como el nombre, descripción, tipo, equipo o reunión.
<b>Resultado Esperado</b>	El proyecto es creado con toda la información actualizada en la base de datos.
<b>Prueba</b>	Correcta.

**Tabla 18: Prueba RF8**

<b>Prueba RF9</b>	<b>Dar de alta una reunión en un proyecto</b>
<b>Descripción</b>	Esta prueba se encarga de comprobar que el administrador o líder de proyecto da de alta una reunión en un proyecto con su descripción y fecha de inicio y fin.
<b>Entrada</b>	Los datos de la reunión y el proyecto asociado.
<b>Resultado Esperado</b>	La reunión es dada de alta, asociándose al proyecto seleccionado.
<b>Prueba</b>	Correcta.

**Tabla 19: Prueba RF9**

### **5.3 Pruebas Integración**

Consiste en la integración de las unidades individuales para formar componentes más grandes. Por ejemplo, dos unidades que ya han sido probadas se combinan entre sí, probándose en conjunto.

### **5.4 Pruebas Regresión**

Son las pruebas que se realizan tras realizar cambios en el proyecto, tras realizar cambios en el proyecto, se han realizado estas pruebas para revisar que todo seguía funcionando correctamente y que los cambios no habían generado fallos. Además, han servido para detectar errores no descubiertos previamente.



## 6 Conclusiones y trabajo futuro

---

### 6.1 Conclusiones

Con el desarrollo de este trabajo he podido elaborar por primera vez una aplicación web, aprendiendo a abordar un proyecto por mí mismo de principio a fin. He sido, por primera vez, el único responsable del desarrollo de un proyecto dentro de una empresa.

Me ha permitido analizar cada fase del ciclo del proyecto sin ponerme desde el principio solamente a implementar, sino también haciendo los diagramas y analizando los requisitos pedidos por la empresa.

He sido capaz de poner en práctica muchos de los conocimientos adquiridos durante la carrera que, ahora soy consciente, han sido una buena base. Para la memoria destacar los conocimientos adquiridos en Informática y sociedad e ingeniería del software, para la parte del desarrollo todas las asignaturas de programación como Programación I, Programación II, Análisis y diseño de Software y Sistemas Informáticos I y para la parte de base de datos la asignatura de Estructuras de Datos.

Además de lo anteriormente citado, he tenido que formarme en otras tecnologías que me han sido exigidas durante el proyecto.

Me gustaría destacar el aprendizaje y soltura adquiridos con el framework .NET y con todos los lenguajes utilizados en el desarrollo, como C#, TSQL, Bootstrap, AJAX, HTML5 Y CSS, poniéndome a prueba cada día del desarrollo, solucionando partes en las que me quedaba estancado.

Una vez terminado el Gestor de Proyectos, este cumple con los requisitos expuestos y cuenta con una interfaz fácil de utilizar y estética, que facilita la gestión de proyectos de la empresa permitiendo mantener un seguimiento del trabajo de un grupo de trabajadores.

### 6.2 Trabajo futuro

Pese a ser una aplicación fácil de usar y tener una interfaz agradable, se seguirá manteniendo y mejorando continuamente con nuevos requisitos. De entre alguna de las posibles mejoras para la aplicación, cabe destacar:

- Mejorar el diseño en otros dispositivos ya que a pesar de usar Bootstrap se pueden optimizar mejor algunos detalles.
- Sería útil un chat en vivo como tiene la herramienta Basecamp y no solo un sistema de mensajería.
- Mejorar las estadísticas realizadas por la aplicación generando informes más elaborados con más información. Si es posible, en un futuro, se podría integrar algún módulo de predicción, que informe con advertencias y recomendaciones, que permita completar un proyecto en el tiempo estimado reduciendo el desfase con el tiempo real lo máximo posible.

- Añadir nuevas medidas de seguridad. Por ejemplo, para proteger la contraseña de un usuario.

# Referencias

---

- [1] Kanban Flow [Última visita 20/02/2019]:  
<https://kanbanflow.com>
- [2] Wrike [Última visita 20/02/2019]:  
<https://www.wrike.com/es/de/>
- [3] Herramientas alternativas [Última visita 20/02/2019]:  
<https://www.eoi.es/blogs/mintecon/2015/01/10/tecnica-o-herramienta-mas-util-en-direccion-de-proyectos-26/>
- [4] Comparativa de las Herramientas [Última visita 20/02/2019]:  
<https://www.ionos.es/digitalguide/correo-electronico/cuestiones-tecnicas/comparativa-de-herramientas-de-gestion-de-proyectos/>
- [5] Modelo-Vista-Controlador [Última visita 24/02/2019]:  
<https://www.tutorialesprogramacionya.com/aspnetmvcya/detalleconcepto.php?punto=2&codigo=2&inicio=0>
- [6] Framework .NET [Última visita 28/02/2019]:  
[https://es.wikipedia.org/wiki/Microsoft\\_.NET](https://es.wikipedia.org/wiki/Microsoft_.NET)
- [7] .NET vs PHP [Última visita 28/02/2019]:  
<https://guiadev.com/php-vs-asp-net/>
- [8] Bootstrap [Última visita 28/02/2019]:  
[https://es.wikipedia.org/wiki/Bootstrap\\_\(framework\)](https://es.wikipedia.org/wiki/Bootstrap_(framework))
- [9] jQuery [Última visita 28/02/2019]:  
<https://jquery.com>
- [10] AJAX [Última visita 28/02/2019]:  
<https://es.wikipedia.org/wiki/AJAX>
- [11] Modelo de ciclo de vida en cascada [Última visita 28/02/2019]:  
<https://www.obs-edu.com/es/blog-project-management/metodologia-agile/pros-y-contras-de-la-metodologia-en-cascada>
- [12] Procedimientos almacenados vs LINQ [Última visita 28/02/2019]:  
<https://www.iteramos.com/pregunta/3349/linq-to-sql-vs-procedimientos-almacenados>
- [13] VMWare Horizon Client [Última visita 28/02/2019]:  
<https://www.vmware.com/es/products/horizon.html>
- [14] Requisitos y tipos [Última visita 28/02/2019]:  
<https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales>
- [15] Virtualización de escritorio [Última visita 28/02/2019]:  
[https://es.wikipedia.org/wiki/Virtualización\\_de\\_escritorio](https://es.wikipedia.org/wiki/Virtualización_de_escritorio)
- [16] Tipos de pruebas [Última visita 28/02/2019]:  
<http://ing-sw.blogspot.com/2005/04/tipos-de-pruebas-de-software.html>
- [17] Microsoft Visual Studio [Última visita 28/02/2019]:  
<https://visualstudio.microsoft.com/es/>
- [18] Microsoft SQL Server Management Studio 17 [Última visita 28/02/2019]:  
[https://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://es.wikipedia.org/wiki/Microsoft_SQL_Server)
- [19] Business Intelligence Development Studio [Última visita 28/02/2019]:  
[https://en.wikipedia.org/wiki/Business\\_Intelligence\\_Development\\_Studio](https://en.wikipedia.org/wiki/Business_Intelligence_Development_Studio)
- [20] HTML [Última visita 28/02/2019]:  
<https://es.wikipedia.org/wiki/HTML>

- [21] CSS [Última visita 28/02/2019]:  
[https://es.wikipedia.org/wiki/Hoja\\_de\\_estilos\\_en\\_cascada](https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada)
- [22] LESS [Última visita 28/02/2019]:  
[https://es.wikipedia.org/wiki/LESS\\_\(lenguaje\\_de\\_hojas\\_de\\_estilo\)](https://es.wikipedia.org/wiki/LESS_(lenguaje_de_hojas_de_estilo))
- [23] Transact-SQL y funciones [Última visita 28/02/2019]:  
[https://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://es.wikipedia.org/wiki/Microsoft_SQL_Server)
- [24] Procedimientos almacenados [Última visita 28/02/2019]:  
[https://es.wikipedia.org/wiki/Procedimiento\\_almacenado](https://es.wikipedia.org/wiki/Procedimiento_almacenado)
- [25] Lenguaje C# [Última visita 02/03/2019]:  
[https://es.wikipedia.org/wiki/C\\_Sharp](https://es.wikipedia.org/wiki/C_Sharp)
- [26] Lenguaje Integrated Query (LINQ) [Última visita 02/03/2019]:  
[https://es.wikipedia.org/wiki/Language\\_Integrated\\_Query](https://es.wikipedia.org/wiki/Language_Integrated_Query)
- [27] Angular, Vue y React [Última visita 10/03/2019]:  
<https://www.uno-de-piera.com/angular-reactjs-vuejs/>

## Glosario

---

<b>API</b>	Application Programming Interface
<b>AJAX</b>	Técnica de desarrollo web.
<b>Aplicación Web</b>	Herramientas con la que los usuarios acceden a un servidor web mediante internet.
<b>Base de datos</b>	Conjunto de datos organizada y almacenados para su uso.
<b>BBDD relacional</b>	Conjunto de datos estructurados que cumplen con el modelo relacional.
<b>BBDD no relacional</b>	Conjunto de datos sin estructura fija que no cumple con el modelo relacional.
<b>Bootstrap</b>	Biblioteca o herramientas para el diseño de aplicaciones Web.
<b>Cliente</b>	Aplicación informática que consume un servicio de otro ordenador.
<b>CSS</b>	“Cascading Style Sheets”. Lenguaje para definir y crear estilos de un documento HTML.
<b>Framework</b>	Entorno de trabajo formado por un conjunto de artefactos y módulos definidos que sirven para el desarrollo software.
<b>GRID</b>	Representación de datos en tablas formadas por filas y columnas.
<b>Hardware</b>	Elementos físicos que componen un Sistema Informático.
<b>HTML</b>	Hyper Text Markup Language. Lenguaje de marcado para la elaboración de páginas web basado en etiquetas.
<b>IDE</b>	Entorno de desarrollo.
<b>JavaScript</b>	Lenguaje de programación interpretado, basado en prototipos, imperativo, débilmente tipado y dinámico.
<b>JQuery</b>	Biblioteca multiplataforma de JavaScript que simplifica la manera de interactuar con los documentos HTML.
<b>JSON</b>	“JavaScript Object Notation”. Formato de texto sencillo para el intercambio de datos.
<b>Librería</b>	Módulo que permite la implementación de una funcionalidad concreta.

<b>Login</b>	Proceso mediante el cual se controla el acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales provistas por el usuario.
<b>MVC</b>	Modelo-Vista-Controlador. Patrón de arquitectura software que separa los datos y la lógica de una aplicación.
<b>Plugin</b>	Aplicación que se relaciona con otra para agregarla una nueva función generalmente específica.
<b>Responsive</b>	Diseño web adaptable para el desarrollo cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo utilizado para visitarlas.
<b>Servidor</b>	Aplicación en ejecución capaz de atender las peticiones de un cliente y darle una respuesta.
<b>Sistema Informático</b>	Sistema que permite almacenar y procesar información.
<b>Software</b>	Componentes lógicos que permiten la realización de tareas.
<b>Stream</b>	Distribución digital de contenido multimedia a través de una red de computadoras.
<b>Tablet</b>	Computadora portátil de mayor tamaño que un teléfono inteligente, integrada en una pantalla táctil con la que se interactúa primariamente con los dedos.

## Anexos

---

### A Anexo

#### i. Anexo Especificación de requisitos

RF9	Dar de alta una reunión en un proyecto
Descripción	La aplicación permitirá al administrador y al líder del proyecto dar de alta una nueva reunión.
Entrada	Los datos de la reunión y el proyecto asociado.
Proceso	El sistema comprobará los datos introducidos y añadirá al proyecto la nueva reunión.
Salida	El sistema informará si se ha dado de alta la reunión con éxito o informando del error en caso de fallo.

**Tabla 20: Requisito funcional 9**

RF10	Modificar información de una reunión
Descripción	La aplicación permitirá al administrador y al líder del proyecto modificar la reunión.
Entrada	El administrador o líder del proyecto introducirá los datos a modificar.
Proceso	El sistema comprobará los datos introducidos y si son correctos modificará la información de la reunión.
Salida	El sistema informará si la modificación de la reunión se ha realizado correctamente o error en caso contrario.

**Tabla 21: Requisito funcional 10**

RF11	Seleccionar líder del equipo del proyecto
Descripción	La aplicación obligará a elegir un líder de proyecto.
Entrada	Usuario líder del proyecto y proyecto.
Proceso	El sistema comprobará que el usuario pertenece al proyecto y es asignado como líder en la base de datos.
Salida	El sistema informará si se ha realizado con éxito o con un mensaje de error si ha ocurrido algún error.

**Tabla 22: Requisito funcional 11**

RF12	Búsqueda de un proyecto
Descripción	La aplicación permitirá al administrador la búsqueda de proyectos por nombre y su tipo.
Entrada	Atributos mencionados en la descripción por los que se quieran filtrar.
Proceso	El sistema filtrará los proyectos que coincidan con las especificaciones y los listará.
Salida	El sistema mostrará la lista de los proyectos que cumplen los filtros.

**Tabla 23: Requisito funcional 12**

RF13	Búsqueda de una tarea
Descripción	La aplicación permitirá al administrador la búsqueda de tareas por nombre y proyecto al que pertenece.
Entrada	Atributos mencionados en la descripción por los que se quieran filtrar.
Proceso	El sistema buscará las tareas que coincidan con las especificaciones y los listará.
Salida	El sistema mostrará la lista de los eventos que cumplen las condiciones.

**Tabla 24: Requisito funcional 13**



<b>RF14</b>	<b>Gráfico Usuario-Proyecto</b>
<b>Descripción</b>	La aplicación permitirá al administrador ver una gráfica del usuario en uno de sus proyectos. Mostrará el porcentaje de desarrollo suyo con respecto al del resto del equipo.
<b>Entrada</b>	Usuario y nombre del proyecto.
<b>Proceso</b>	El sistema comprobará las estadísticas.
<b>Salida</b>	El sistema mostrará el grafico resumen del proyecto.

**Tabla 25: Requisito funcional 14**

<b>RF15</b>	<b>Gráfico Diario-Usuario</b>
<b>Descripción</b>	La aplicación permitirá al administrador ver una gráfica del usuario en uno de sus proyectos en un día determinado. Mostrará su trabajo total en dicho proyecto, en el resto de proyectos y en la resolución de incidencias.
<b>Entrada</b>	Usuario y nombre del proyecto y día.
<b>Proceso</b>	El sistema comprobará las estadísticas.
<b>Salida</b>	El sistema mostrará el grafico resumen.

**Tabla 26: Requisito funcional 15**

## ii. Anexo Pruebas Funcionales

<b>Prueba RF17</b>	<b>Dar de baja un proyecto</b>
<b>Descripción</b>	La aplicación permitirá al administrador dar de baja a un proyecto.
<b>Entrada</b>	El administrador deberá introducir el nombre del proyecto.
<b>Resultado Esperado</b>	El proyecto es eliminado de la base de datos.
<b>Prueba</b>	Correcta.

**Tabla 27: Prueba RF17**

<b>Prueba RF18</b>	<b>Dar de baja un usuario en un proyecto</b>
<b>Descripción</b>	Esta prueba se encarga de comprobar que el administrador o líder de proyecto puede dar de baja a un usuario en el equipo del proyecto.
<b>Entrada</b>	Nombre de usuario y proyecto.
<b>Resultado Esperado</b>	El usuario se elimina del proyecto correctamente.
<b>Prueba</b>	Correcta.

**Tabla 28: Prueba RF18**

<b>Prueba RF19</b>	<b>Eliminar una tarea</b>
<b>Descripción</b>	La aplicación permitirá eliminar una tarea al administrador, líder de proyecto o propietario de la tarea.
<b>Entrada</b>	El administrador, líder de proyecto o propietario de la tarea deberá introducir el nombre de la tarea y el proyecto.
<b>Resultado Esperado</b>	La tarea es eliminada de la base de datos.
<b>Prueba</b>	Correcta.

**Tabla 29: Prueba RF19**

<b>Prueba RF10</b>	<b>Modificar información de una reunión</b>
<b>Descripción</b>	Prueba que permite al administrador y al líder del proyecto modificar una reunión.
<b>Entrada</b>	El administrador o al líder del proyecto introducirá los datos a modificar.
<b>Resultado Esperado</b>	La reunión es creada con toda la información actualizada en la base de datos.
<b>Prueba</b>	Correcta.

**Tabla 30: Prueba RF10**

<b>Prueba RF11</b>	<b>Seleccionar líder del equipo del proyecto</b>
<b>Descripción</b>	Esta prueba se encarga de comprobar que el líder de proyecto se ha asignado correctamente.
<b>Entrada</b>	Usuario líder del proyecto y proyecto.
<b>Resultado Esperado</b>	El líder se asigna correctamente en dicho equipo del proyecto.
<b>Prueba</b>	Correcta.

**Tabla 31: Prueba RF11**

<b>Prueba RF12</b>	<b>Búsqueda de un proyecto</b>
<b>Descripción</b>	Prueba que permite la búsqueda de proyectos por nombre y su tipo.
<b>Entrada</b>	Atributos mencionados en la descripción por los que se quieran filtrar.
<b>Resultado Esperado</b>	El proyecto es filtrado mostrándose en pantalla.
<b>Prueba</b>	Correcta.

**Tabla 32: Prueba RF12**

<b>Prueba RF13</b>	<b>Búsqueda de una tarea</b>
<b>Descripción</b>	Prueba que permite la búsqueda de tareas por nombre y proyecto al que pertenece.
<b>Entrada</b>	Atributos mencionados en la descripción por los que se quieran filtrar.
<b>Resultado Esperado</b>	La tarea es filtrada mostrándose en pantalla.
<b>Prueba</b>	Correcta.

**Tabla 33: Prueba RF13**

<b>Prueba RF14</b>	<b>Gráfico Usuario-Proyecto</b>
<b>Descripción</b>	La aplicación permitirá al administrador ver una gráfica del usuario en uno de sus proyectos. Mostrará el porcentaje de desarrollo suyo con respecto al del resto del equipo.
<b>Entrada</b>	Usuario y nombre del proyecto.
<b>Resultado Esperado</b>	Gráfica resumen.
<b>Prueba</b>	Correcta.

**Tabla 34: Prueba RF14**

<b>Prueba RF15</b>	<b>Gráfico Diario-Usuario</b>
<b>Descripción</b>	La aplicación permitirá al administrador ver una gráfica del usuario en uno de sus proyectos en un día determinado. Mostrará su trabajo total en dicho proyecto, en el resto de proyectos y en la resolución de incidencias.
<b>Entrada</b>	Usuario, nombre del proyecto y día.
<b>Resultado Esperado</b>	Gráfica resumen.
<b>Prueba</b>	Correcta.

**Tabla 35: Prueba RF15**

### iii. Ficheros de configuración

**En el WebConfig** se realiza la conexión con la base de datos del Gestor de Proyectos.

```
<connectionStrings>
  <add name="Miguel" connectionString="data source =***.***.**.***; User
ID=sa;Password=*****;Integrated
Security=True;Database=GestorProyectos;Trusted_Connection=False"
providerName="System.Data.SqlClient"/>
</connectionStrings>
```

**En el Route Config** se indica la vista donde comienza el gestor de proyectos (inicialmente, se utilizó el login como vista inicial para el TFG, sin embargo, en la empresa comienza en el menú principal del gestor ya que se usa un login global para acceder a la intranet y una vez dentro de esta, dependiendo de tu perfil podrás acceder a algunas aplicaciones, entre ellas el Gestor de Proyectos en la que una vez entras, ya estarás logueado con tu usuario).

```
public class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

        routes.MapRoute(
            name: "Default",
            url: "{controller}/{action}/{id}",
            defaults: new { controller = "Vista", action = "Login", id = UrlParameter.Optional
        }
    );
    }
}
```

**BundleConfig:** Los bundles es el mecanismo que tiene ASP.NET MVC para incluir varios ficheros (de script o css) que están relacionados entre ellos.

En el archivo App\_Start/BundleConfig tenemos:

```
bundles.Add(new ScriptBundle("~/bundles/bootstrap").Include(
    "~/Scripts/bootstrap.js",
    "~/Scripts/respond.js"));
```

En este método, que es llamado desde el Application\_Start, es donde se asocia cada bundle con un conjunto de archivos.

Con el uso de los bundles se puede agrupar lógicamente nuestros archivos javascript o css.

#### iv. Organización de las carpetas

En esta sección se expone la organización del Gestor de Proyectos explicando sus componentes más importantes.

- La carpeta más importante es el **GestorProyectos** (donde se encuentra la aplicación). En esta carpeta se localizan los controladores, los modelos, las vistas y los archivos de configuración.
- La carpeta **public** contiene archivos como los plugins y las hojas de estilo entre otros.
- La carpeta **packages** contiene todas las librerías.
- La carpeta **config** contiene los archivos de configuración de la aplicación.
- La carpeta **model** contiene los modelos de la aplicación.
- La carpeta **views** contiene las vistas de la aplicación.
- La carpeta **controller** contiene los controladores de la aplicación.